

# Middlesex University Research Repository

An open access repository of

Middlesex University research

<http://eprints.mdx.ac.uk>

Schofield, Simon (1994) Non-photorealistic rendering: a critical examination and proposed system. PhD thesis, Middlesex University. [Thesis]

This version is available at: <https://eprints.mdx.ac.uk/6723/>

## Copyright:

Middlesex University Research Repository makes the University's research available electronically.

Copyright and moral rights to this work are retained by the author and/or other copyright owners unless otherwise stated. The work is supplied on the understanding that any use for commercial gain is strictly forbidden. A copy may be downloaded for personal, non-commercial, research or study without prior permission and without charge.

Works, including theses and research projects, may not be reproduced in any format or medium, or extensive quotations taken from them, or their content changed in any way, without first obtaining permission in writing from the copyright holder(s). They may not be sold or exploited commercially in any format or medium without the prior written permission of the copyright holder(s).

Full bibliographic details must be given when referring to, or quoting from full items including the author's name, the title of the work, publication details where relevant (place, publisher, date), pagination, and for theses or dissertations the awarding institution, the degree type awarded, and the date of the award.

If you believe that any material held in the repository infringes copyright law, please contact the Repository Team at Middlesex University via the following email address:

[eprints@mdx.ac.uk](mailto:eprints@mdx.ac.uk)

The item will be removed from the repository while any claim is being investigated.

See also repository copyright: re-use policy: <http://eprints.mdx.ac.uk/policies.html#copy>

# **Non-Photorealistic Rendering:**

**A critical examination and  
proposed system**

**Simon Schofield**

**A dissertation submitted for the degree of Doctor of Philosophy  
Middlesex University  
March 1994**

# Table of Contents

	<i>Page</i>
Acknowledgements and Declarations	1
Summary of programme	2
Keywords	3
 Chapter 1 – Introduction	 4
Chapter 2 – Photorealism and Non-photorealism	6
2.1 – The Dominance of Photorealism	
2.2 – Photorealism and Truth	7
2.3 – Other forms of representation may be more apt	9
2.4 – Aesthetic criticisms of Photoreal Computer Graphics	11
2.5 – The emergence of Non Photorealistic rendering	12
 Chapter 3 – A review of previous systems	 14
3.1 – Mezei et. al.	
3.2 – Yessio	
3.3 – Strassman	
3.4 – Sasada	15
3.5 – van Berkal	16
3.6 – Barnsley et. al.	
3.7 – Haeberli	17
3.8 – Paint systems	19
3.9 – Cockshott	20
3.10 – Saito & Takahishi	21
3.11 – Freehand Plotting	22
3.12 – The KATI renderer	23
3.13 – Summary and conclusion	24
 Chapter 4 – Criticisms of Current Approaches	 27
4.1 – The name "Non-photorealistic Rendering"	
4.2 – Why NPR is abhorrent to some	28
4.3 – Meaningless effects leading to trivial images	
4.4 – Generating more significant images	31
4.5 – Traditional techniques for generating significance	32
4.6 – Image-space vs. World-space	35
4.7 – Drawing as a hierarchical process	37
4.8 – Summary of techniques for improving quality	39
 Chapter 5 – Mature NPR is not painting simulation	 40
5.1 – The comparison of NPR to real painting and drawing	
5.2 – Avoiding accusations of simulation through use of the transcendental qualities of hand-media	43
5.3 – Summary of prescriptions for mature NPR	46

<b>Chapter 6 – NPR's relationship to other media</b>	<i>Page</i> 49
<b>Chapter 7 – Developing an NPR system</b>	51
7.1 – Recap of Reccomendadtions made to-date	
7.2 – Interactivity vs. Batch rendering	52
7.3 – Consultations with Architects and Designers	54
7.4 – Benefits of NPR	56
7.5 – A wish-list for an NPR system	60
7.6 – Application overview	61
<b>Chapter 8 – The Expressive Marks System</b>	63
8.1 – Introduction to the implementation	
8.2 – The Mark Synthesiser	
8.3 – Marks used to post-process images and Magic Painter	70
8.4 – Marks used to render geometries	73
8.5 – Other Graphic elements	76
<b>Chapter 9 – The Enriched-Pixel Scene Description</b>	78
9.1 – The EPixel approach to rendering scenes	
9.2 – Geometric methods using EPixel images	82
<b>Chapter 10 – EPixel based Advanced NPR techniques</b>	88
10.1 – Point distribution and Line orientation	
10.2 – Drawing more complex entities	91
<b>Chapter 11 – An OO approach to defining and applying NPR techniques</b>	94
11.1 – The Sketcher system	
11.2 – Control flow	97
11.3 – Sketcher Nodes and other elements	98
11.4 – Combining the elements in the Piranesi system	103
11.5 – Summary	
<b>Conclusions</b>	
12.1 – Results and contribution to knowledge	105
<b>References and Bibliography</b>	109-114



# **Non-photorealistic Rendering: A critical examination and proposed system**

by Simon Schofield B.A. M.A.

being a thesis submitted in support of the degree of Doctor of Philosophy,  
School of Art and Design, Middlesex University, October 1993

## **Acknowledgements**

I would like to thank the following people and organisations for making this investigation both possible and rewarding. From Middlesex University, those members of CASCAAD, who gave me help and programming advice in the early days and to Professor John Lansdown and Dr. Gregg Moore for their advice, enthusiasm and encouragement throughout the project. Noble Campion Ltd who provided generous sponsorship while at Middlesex. Members of the Martin Centre, University of Cambridge, particularly Paul Richens, John Rees and Tim Wiegand, who's expertise and patient help was indispensable. Informatix Inc (Tokyo) for providing generous funding for the establishment of the Martin Centre CADLab. Many architectural practices who have contributed their opinions, enthusiasm and models, especially Miller•Hare (John Hare and Steve Miller), Nicholas Ray Associates (Nick Ray), Sir Norman Foster and Partners (Caroline Brown), Scott Brownrigg and Turner (Roger Mathews).

## **Declaration of Originality**

The material presented in this thesis is completely my own work carried out at CASCAAD, Middlesex University and at the Martin Centre CADLab, University of Cambridge. Where ever secondary material has been used by me, either published or unpublished, full references and acknowledgements are given.

*Simon Schofield 1993*

# Summary of Programme

In the first part of the program the emergent field of Non-Photorealistic Rendering is explored from a cultural perspective. This is to establish a clear understanding of what Non-Photorealistic Rendering (NPR) ought to be in its mature form in order to provide goals and an overall infrastructure for future development. This thesis claims that unless we understand and clarify NPR's relationship with other media (photography, photorealistic computer graphics and traditional media) we will continue to manufacture "new solutions" to computer based imaging which are confused and naive in their goals. Such solutions will be rejected by the art and design community, generally condemned as novelties of little cultural worth ( i.e. they will not sell).

This is achieved by critically reviewing published systems that are naively described as Non-photorealistic or "painterly" systems. Current practices and techniques are criticised in terms of their low ability to articulate meaning in images; solutions to this problem are given. A further argument claims that NPR, while being similar to traditional "natural media" techniques in certain aspects, is fundamentally different in other ways. This similarity has lead NPR to be sometimes proposed as "painting simulation" – something it can never be. Methods for avoiding this position are proposed. The similarities and differences to painting and drawing are presented and NPR's relationship to its other counterpart, Photorealistic Rendering (PR), is then delineated. It is shown that NPR is paradigmatically different to other forms of representation – i.e. it is not an "effect", but rather something basically different.

The benefits of NPR in its mature form are discussed in the context of Architectural Representation and Design in general. This is done in conjunction with consultations with designers and architects. From this consultation a "wish-list" of capabilities is compiled by way of a requirements capture for a proposed system.

A series of computer-based experiments resulting in the systems "Expressive Marks" and "Magic Painter" are carried out; these practical experiments add further understanding to the problems of NPR. The exploration concludes with a prototype system "Piranesi" which is submitted as a good overall solution to the problem of NPR. In support of this written thesis are :-

- The Expressive Marks system
- Magic Painter system
- The Piranesi system (which includes the EPixel and Sketcher systems)
- A large portfolio of images generated throughout the exploration

# Key Words

Computer Graphics

Visual Representation

Non-photorealistic Rendering

Natural Media Simulations

Rendering

Post-processing

# Chapter 1

## Introduction

This thesis is an analysis of current Computer Graphic Non-Photorealistic Rendering (NPR) and a prescription for future development, culminating in an actual prototype system. It is written from the viewpoint of a Painter and Designer, rather than a Computer Scientist.

Although the term "Non-Photorealistic rendering" is becoming a standard phrase among Computer Graphics (CG) practitioners it remains extremely badly defined. The 1990 SIGGRAPH conference devoted an entire section of its papers to this subject [SIGGRAPH 90] but made no attempt at an overall clarification of what constituted non-photorealistic rendering, what it looks like or what it aims to do. In fact the field is almost entirely without infrastructure: there are no standard texts nor, when all is considered, are there any examples of software fully matching the aspirations of the concept.

Non-photorealistic rendering is, broadly speaking, any type of computer graphic imagery which tries to get away from the paradigm of photorealism. For historical reasons the default tendency of such systems is to generate imagery which is, in some way, akin to those made by hand – e.g. pen and wash, water-colour, charcoal drawing. To this end several systems have been presented, usually only in prototype form [Strassman 85], [Haeberli 90]. Systems which have found their way into commercial use are effectively image-processors [Fractal Design 91]. However it is felt that the most interesting developments will emerge when the systems are, in a functional sense, much more like complete 3-D renderers – capable of interpreting all the data available in a 3-D model in order to generate images which possess more internal meaning and structure. At least one solution based on this principal is, at the time of writing, near commercial release [KATI 93].

This thesis begins with a discussion of the cultural and historical factors surrounding the field and reviews previous relevant papers and systems. The purpose of this overtly discursive part is to redress my point that, as yet, the field is undefined. It aims at both tightening the area of concern and at presenting good goals for non-photorealistic rendering systems. As such the overall objective is to provide a strong cultural infrastructure for future Non-Photorealistic Rendering development.

It will be seen that the most significant problems associated with Non-Photorealistic Rendering are primarily aesthetic and only secondarily technical – aesthetic desires predicate technical innovation. The culturally oriented theoretical problems surrounding the field are many, and at

their most difficult, broach the central issues of perception and representation. However this is not to say that the technical problems of implementing a system are not difficult to overcome. No general prescribed framework has been published for an NPR system, only an array of more focused solutions. And while many of these solutions present useful methods for non-photorealistic techniques, there are still many specific problems to be tackled.

Following the culturally oriented prescriptions made in the first section, and a requirements specification for a system based on a consultation with designers and architects, a detailed description of a series of experiments into NPR is provided.

Along the way several interim prototype systems have been written (Expressive Marks, Magic Painter). The culmination of the experiments is the system "Piranesi", which meets the overall prescriptions made in the earlier sections. These software solutions and the images thereby produced form an important part of the Ph.D. programme and ought to be considered "the results" of the investigation.

## Chapter 2

# Photorealism and Non-photorealism

The emergence of non-photorealistic rendering as a desire, concept and possibility is discussed here in its historical context; i.e. as a reaction to prevailing photorealistic modes of representation.

### 2.1 The Dominance of Photorealism

Over the last few decades most research into high resolution computer graphics, particularly that concerned with representing three-dimensional worlds, has concentrated on generating photorealistic images. The reasons for this mainly relate to CG's historical associations with a scientific community, rather than an artistic community.

Primarily, Photorealism enjoys a privileged position amongst the universe of images. This is due to a long historical struggle to represent the world with accuracy to its material/physical – i.e. objective – appearance. As Barthes points out the addition of the chemical element of Photography in the early 19th century, while often cited as the beginning of photography proper, was only another addition to a long line of known photorealistic techniques developed from the early Renaissance onwards [Barthes 71]. Use of the *camera obscura*, tracing shadows and projections through meshes, sheets of glass and linen as well as a firm grasp of the mathematical laws of perspective were all common techniques long before the first permanent, chemical photograph in 1826. The consuming nature of this struggle, and its attendant successes, has led many people to unquestioningly regard photorealism as "more correct" than other forms of representation, and therefore more desirable. This position is particularly defended by the Materialist View of the universe [Sorokin 92] which presumes that ultimate truth is underpinned by an underlying physical/objective reality which can be unearthed through rational investigative processes – i.e. scientific discovery. Because photorealism is the result of understood optical processes, which are part of the physical/objective universe (i.e. not mediated by a "fallible" human interlocutor) it is seen by the Materialists as closer to reality than other forms of human-mediated representation. Because most practitioners and researchers concerned with CG have historically worked within these Materialist, objective, positivist paradigms it is not surprising that photorealism is seen as the ultimate pictorial goal.

To those working within a scientific framework, photorealism is not only seen as highly desirable

but also highly achievable. The understanding of photorealism as "objective" is the result of literally centuries of investigation into the laws of optics and perspective. The application of perspective laws, light, material properties and lenses significantly predate the appearance of the computer as the supporting media. These insights are now being turned into algorithms for generating synthetic photographs of virtual realities.

However, when viewed from a broader cultural perspective photorealism is, of course, only one pictorial style from many. It can be seen that these other styles are not used in want of photorealism but are actually used in preference for whatever reason. The use of alternative and radical forms of representations is particularly strong in the areas of Art and Design where it is felt that there are several reasons why PR may sometimes be less than desirable. These are now discussed in detail.

## 2.2 Photorealism and Truth

The dominance of photorealism and its arrogant claims toward objectivity and correctness has engendered a running counter battle from those who suspect photorealism as being ultimately-coercive and so potentially truth-hiding, rather than truth-revealing. Notably, Ernst Gombrich is irritated to find himself in a world dominated by an unquestioning regard for the sanctity of the photograph. He takes it upon himself to undermine the presumption that photography is a correct or "direct" form of representation – that it is more intimately bonded to the reality it represents than other forms of images. He does so by showing how heavily abstracted a photograph actually is through exposing the many artifices employed – the micro instant frozen forever, the limited angle of view and the arbitrariness of photographic processing. [Gombrich 60].

These anxieties have recently been fanned by the arrival of photorealistic computer graphics and particularly by the immanent arrival of Virtual Reality.

*What they seek is a kind of 'realism' of an ideal sort, a realism that tries to describe the world with an insistent, even authoritarian, accuracy that is overwhelming. It is as though the corporate power of the media had joined forces with the methodological rigour of the mathematicians and scientists to create some final, definitive and coercive depiction of the visual world.*

[Wright 89]

Virtual Reality manifests our cultural imagination's extreme in terms of a functioning and completely convincing seduction by the unreal. It can be seen as embodying a property which may be termed "consensual realism" [Rheingold 91]; that it seeks to seduce the user through a type of

*trompe l'oeil* into believing that the thing represented is actually present. In order to achieve this *trompe l'oeil* the mediating mechanisms of representation must hide themselves in order to convince that it is NOT an image. In order to secure this seduction it is often proffered that the artifice of illusion or, more specifically, representation must become invisible or transparent. For this reason any technique enabling the production of photorealistic imagery is seen as an agent in an enterprise of coercion, seduction and, ultimately, deception.

Several more recent theorists have adopted a highly aggressive position against the overbearing invasion of photographic or hyper-real imagery. Jean Baudrillard states the trend towards complete artificial realism is an enterprise of mass deception which will lead to an irrevocable loss of power to interpret what we see, that the total consensual realism offered by Virtual Reality essentially constitutes the site of loss of meaning and understanding [Baudrillard 87] rather than its supposed consolidation. His famous essay *The Procession of Simulacra* taken from his larger thesis *Simulations* [Baudrillard 83] is often referenced in relation to computer graphic photorealistic imagery because it warns of our ability to create things that seem real but are not, and our ability to then accredit them with qualities and merits they do not deserve. The urge to understand through sheer looking (the gaze) constitutes a fatal trajectory for scientific investigation to pursue because we may find ourselves believing simulations to be "true" merely because they look like photographs, rather than have other, deeper "truth values" [D. Carroll 87][Leyotard 84]. Photorealism and particularly artificial photorealism, thus, may be seen as the agent for seduction into a state of meaningless inter-referentiality, wherein we suspend our disbelief of the false.

On a more pragmatic note, Douglas Crimp [84] worries that, because we are too happy to accept the photograph as a surrogate for the real, it will be used to replace the real when the real becomes too valuable or difficult to display, leading to a further impoverishing of our permitted experience.

An inference of this argument is that a medium whose mechanisms are inscrutable is somehow dishonest. Those representational processes which, in the final generated image, admit their presence are securing that the image is not going to be mistaken for something else (e.g. the "real"). This view is supported by Linda Nochlin

*There is a tendency in illusionist art - art that stresses the existence of credible 3D forms in a believable space - to minimise the pictorial means, to make them disappear... and in the 19th C accusations of 'deception' and 'dishonesty' were hurled at those, who through foreshortening, over subtle modelling or minute description of surfaces, would tend to make mock of the recalcitrant impermiability of the pictorial surface.*

[Nochlin71]



As such, more self-admitting traits seen in certain types of images lead us to regard them as being more honest by admitting themselves to be "just images". It is extremely difficult to disguise the techniques and process of painting and drawing, and so many paintings and drawings are often strong examples of this type of "honest" imagery.

## 2.3 Other forms of Representation may be more apt

Despite these repeated warnings as to the seductive and coercive nature of photorealism, it is still often regarded as yielding more of the referent – exposing more of it – than other forms of representation. For instance, one would wish to see a photograph of a person in preference to a drawing in order to later recognise that person in an identity parade. One may remember the case of David Hockney attempting to use a photorealistic drawing of himself in his passport, and consequently it being invalidated.

However, it can be quite clearly shown that photorealism is not always the best device for depiction and in some cases would be quite useless. This is crudely demonstrated if we compare the types of information stored in the Bayeux Tapestry to those stored in the mass of photographs of the First World War – one undoubtedly serves historical interpretation in a very different way to the other. Given both, without supporting text, the historian could only infer so much from the photographs; a mass of specifics without an overall coherence. The tableau at least yields a general relatable story through its lack of specificity as well as some actual visual specifics. Or consider the value a photograph of a car engine compared with that of an exploded schematic line drawing – what use would a photograph be to a mechanic when he already has the real thing in front of him? Photographs can perform very badly when clear and precise delineation, explanation and understanding of some actual referent is required. It is not surprising then that other representational methods are used when planning and designing things. The traditional tools of pen and pencil, standard legends and notation, agreed conventions and certain methods of presentation, such as the projections used in building plans serve for many reasons better than any amount of photographic documentation.

One must ask the hypothetical question that if an architect were able to travel forward in time in order to take photographs of a proposed building (to show his client) would he want to? There are two reasons to suggest he would not. Firstly photographs are notoriously bad at communicating buildings in terms of clearly demarking materials, structure and relationships.

*Photographs are undoubtedly very popular but orthographic projections and especially floor-plans and sections remain the main source of information on spatial articulation and building structure ... as they convey aspects which even a visit to the actual building would fail to reveal.*

[Koutamanis 93]

If a visit to the actual building fails to reveal layers of information, then it is doubtful that high resolution simulations would fair any better. One counter-argument claims that photography is not used in the planning and designing of artefacts precisely because, before something is built, there is nothing to photograph and so it follows that other modes of representation must be used as some sort of "second best". However, with the advent of computer simulations very convincing photographs *can* be artificially manufactured. This is now becoming an available technique for the many manufacturing industries including product design, the fashion industry, architecture, and even cosmetic surgery. However, despite this new ability to be able to construct synthetic photographs, this type of imagery is very carefully metered out and treated as though it is, in some way, toxic. This relates to the second reason why the time-travelling architect would not wish to show his client the photograph of the, as yet, unbuilt building. There are significant expectations with which we burden photorealistic images [Bergin 90], [Barthes 71] due to our *belief* in them as documentary of the real.

Photorealism is particularly poor at dealing with the general because it insists on the specificity of the referent – photorealism implicitly insists that the image presents all the detail originally in referent, with minor considerations of focus and grain-size. Hence, a photograph can be inspected under a microscope to reveal further detail if desired. All PR imagery, as a consequence possesses an inbuilt suggestion of *completeness*. We are not used to looking at "partial photography", where the scene is not complete, not all present. To use a rather awkward simile, looking at PR images of uncompleted scenes is rather like looking at a photograph of some people who have not yet got faces and hands – rather disconcerting. This psychological response to PR imagery is not surprising given that our reading of photographs is based on the exclusive experiencing of photographs throughout a period when they were regarded as unquestionable documentary. We must now slough off these presumption if we are to read artificially manufactured photorealistic imagery correctly.

Hence the client, presented with a photograph of the building would be diverted immediately to the specifics of the image, rather than any underpinning concepts relating to the general design. Discussing ideas and developing a design in the face of such conceptually numbing imagery would be all the harder.

## 2.4 Aesthetic criticisms of photorealistic CG

If we look at a selection of CG 3-D photorealistic images, produced over a period as brief as the last five years (1988-93) we are struck by the vast improvements that have been made over that period, and this is to the credit of research in the field. Despite these improvements it is still extremely difficult to wrest visually pleasing images from most rendering systems. Consequently there is often a resultant blandness or sterility seen in many PR CG images.

The generally accepted method for redressing this problem is to try and raise the intricacy of synthetic images until they approach the visual intricacy of natural photography. In some cases this has been achieved; many completely synthetic scenes are able to convince unsuspecting people that they are real photos. Foley and vanDam [89: plate III.19] present two seemingly identical images of a board-room side by side – one is a real photo, the other a rendering. It takes a sharp eye, and an understanding of computer graphics, to be able to tell which is which. They go on to list comprehensively the mainstream techniques used in giving a scene a photorealistic quality, taking a scene from a basic wire-frame through to a highly finished virtual photograph utilising the increasingly available effects of anti-aliasing, multiple light sources, soft shadows, texture, bump, environment and displacement mapping [op cit.: plates II.21-II.37]. These techniques are now integrated into most marketed renderers such as Renderman [Upstil 90]. Other techniques such as simulated depth of field, lens flare and motion blurring can also be added to the scene. Intriguingly, these effects are often unwanted artefacts of classical photography and are yet deliberately introduced into computer graphic scenes in order to convince us of their veracity to the photograph rather than their ability to acquire the unfettered naturalism of a putative virtual world.

Several new modelling techniques also add desired richness to an image by providing means to model complex, fluid or organic forms, for which classical geometric modelling techniques prove inadequate. Soft-modelling, a way of defining runny, “blobby” objects, provides an extremely economical alternative to the usually rigid shapes defined through boundary representation or constructive solid geometry [Wyvill et al. 86]. Particle systems [Reeves 83] render clouds and complex interactions and provide ways of generating fuzzy objects. HyperTextures render scalar field intensities using a ray-tracing approach, by intersecting geometric figures with multiple noise sources resulting in eroded and fuzzy modifications of the geometric shape [Perlin 89]. Iterative Function Systems, an off-shoot of fractal research, proposes that many forms, while ostensibly complex can be synthesised using simple formula in an iterative manner, and has been shown to work for a limited number of forms (clouds, plants, crystals) successfully [Barnsley et al 88]. Generally these novel approaches are presented as ways of increasing image complexity within

the photorealistic paradigm.

Despite these innovations in modelling and rendering techniques their integration into CAD systems in ways that enable easy use is still some way off . After modelling something of enough detail to generate a sufficiently intricate image a user might spend that time again, setting lights, perspectives, applying textures, adjusting material properties and finding view-points in order to generate a pleasing image. As a consequence of these difficulties relatively few 3-D computer generated images produced reach the intricacy or elegance commonly expected from the traditional mediums of photography or painting and drawing. Undoubtedly these specific problems will be redressed in time.

To summarise, despite photorealism's often assumed privileged status as a representational technique, there are several clear reasons why it is, in certain contexts, problematic. Primarily, on a purely pragmatic level, photorealism is only one style from many and other styles may function better in many circumstances for perceptual or cognitive reasons, or are simply more pleasing. On a subtler level, but one which is deeply felt, photorealism is often perceived by the cultural critical mass (including artists and designers) as a "politically incorrect" form of representation; it is too strident, arrogant and coercive. As such photorealism is often perceived as a totalitarian form of representation. In a rather elliptical and unfair way, this view has recently gained ammunition from critics of photorealistic computer graphics imagery in the film and television industry, who have deemed it an ugly and crass form of graphics when compared to more traditional methods such as hand animation or live footage. It now seems that, despite very rapid advances and improvements in the quality of photorealistic graphic techniques, an aesthetic prejudice has been established that will take some time to shake off.

## 2.5 The Emergence of Non Photorealistic rendering

In response to these difficulties and objections relating to photo-realistic (PR) methods of computer image making and representation, several novel approaches have been offered in an attempt to move CG away from photorealism altogether – towards what has simply become called Non-Photorealistic Rendering (NPR) . These new forms of rendering are often proposed to be a fundamental alternative to PR mode of representation and not a subtle effect or nuance added on top of current techniques. Often they are presented as paradigmatically different to PR although the reasons for this conviction have not, as yet, been clearly stated.

If we jump the gun a little and suggest that NPR is like real painting and drawing, in some respects,

then one would expect it to enjoy the benefits of those techniques. Hence, this ought to be a strong motivating factor behind a lot of extant research. Characterisation, abstraction, interpretation, invention and the unknowable subjective poetics of representation are all things which are felt more achievable through the use of pencil and paintbrush than through the camera. However, as I aim to show, to date there is no significant evidence that these capabilities are easily achievable through computer-based rendering techniques, and so cannot be in themselves cited convincingly as reasons for a new direction in image production. They can be stated as vague wishes, but not offered up as part of a requirement specification because, without further work, the objectives are unclear on two counts; what specific techniques ought to be evolved and how they might be implemented if they can be implemented at all. While we can easily avoid photorealism, the more positive aspects of painting and drawing, which seem to be the long term objective, are without clear predicates.

Hence, it may be true to say that NPR is a field borne out of frustration with photorealism rather than in pursuit of some more positive and definite goals. The very name "Non-Photorealistic Rendering" attests to the underlying negative dynamic of the field and is the root of some frustration when it comes to attempting a tighter definition of the area as it only tells us what the thing *is not*. This absence of a positive dynamic has caused the field to be constituted in a very naive and intuitive way.

One important purpose of this thesis then is to define the nature of NPR in its mature (eventual) form, to give it clear goals and to offer ways of achieving those goals. I start my investigation into NPR with a survey of the systems published. The only criteria for inclusion in the survey is that a system aims at providing alternatives to photorealism (provides a new representational technique) or is attempting a similarity to an accepted form of non-photorealism - e.g. painting or drawing or some other traditional technique; that is to say, those systems and publications which fit an initial naive model of NPR. I have included in this list of reviewed NPR systems several which some would argue are not renderers at all, but post processors which utilise painterly effects [Fractal Design 91],[Haeberli 90], [Strassman 85] to re-generate imagery in automatic or semi-automatic ways. With particular reference to Haeberli's work, I would argue that there is no clear formal distinction between rendering and post-processing – many of their associated methods and functionality are the same bar trivial differences in the types of source data; one uses an image and one uses a model. It will be seen throughout this dissertation how the concept of NPR erodes the distinction between post-processing and rendering to the point where no final distinction can be made.

Despite the initial naive and broad classification of NPR this thesis aims to indicate what NPR will be in its mature form and its relationships to CG photorealism and traditional, natural media non-photorealistic techniques. Most importantly, this section aims at providing a set of prescriptions for future development based on a thorough understanding of the potential cultural role of NPR.

## Chapter 3

### Previous Systems

#### 3.1 Mezei, Puzin and Conroy's Texture Elements

Mezei et al. [74] propose a technique for generating images that bare some relation to hand drawn illustrations. The system is initially intended to synthesise patterns seen in nature and it is proposed that this technique may achieve high degrees of natural realism. It can be seen as a forerunner of most current non-photorealistic systems in its application of a mass of pre-figured graphical subunits (larger than the pixel) to form an image. The general thesis is that many things in nature are composed of visually distinct elements arranged in various distributions. These subunits, or "texture elements" can be synthesised and a range of effects achieved by resizing, rotating and generally perturbing them before application to the image [Figure 1]. Their distribution across a surface is controlled by probabilistic algorithms.

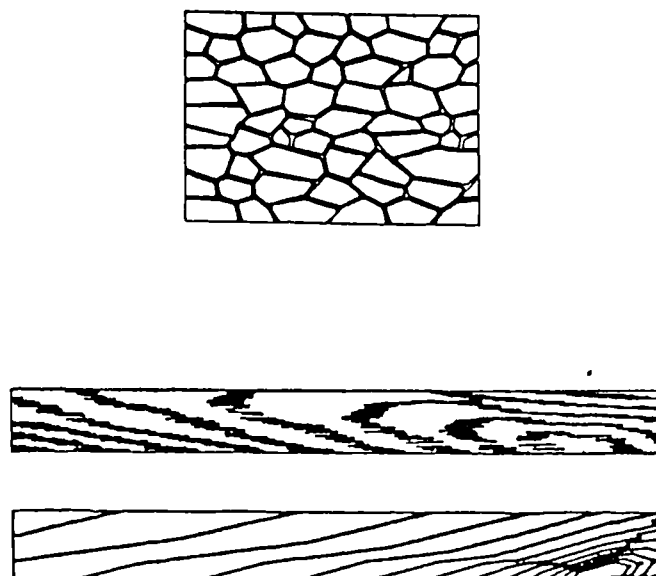
#### 3.2 Yessio's computer drafting techniques

Yessio [79] proposes a range of techniques for a computer drafting system which aim at emulating various standard shading techniques used by draftspeople in technical drawing and illustration with particular attention to rendering the internally complex ordering seen in brick coursing or crazy-paving. A technique is proposed for forming rafts of regular and irregular shapes separated by lines of varying thickness. This is achieved by generating a regular grid which approximates the configuration of the desired material (e.g. a hexagon grid for rubble, a half-dropped rectangular grid for brick) and then perturbing that grid in certain ways. Finally the lines separating the regions are redrawn at random thicknesses. Wood grain effects are achieved by repeatedly perturbing regularly ordered concentric rings [Figure 2].

#### 3.3 Steven Strassman's "Hairy Brushes"



*Fig. 1 Rock, bark and wood grain by Mezei et al.*



*Fig. 2 Crazy Paving, Wood-grain by Yessio*

Strassman's efforts [Strassman 85] stem from his interest in modelling fuzzy, complex systems and focus very much on behavioural simulation of paintbrushes and ink, using object-oriented techniques to create sets of rules and classes of objects which accurately model the way a wet brush works. The elements of the simulation are defined thus:-

*Paint brushes are modelled as a collection of bristles which evolve over the course of the stroke, leaving a realistic image of a Sumie brush stroke. The major representational units are (1) Brush: a compound object composed of bristles, (2) Stroke: a trajectory of position and pressure, (3) Dip: a description of the application of paint to a class of brushes and, (4) Paper: a mapping onto the display device. This modular system allows experimentation with various stochastic models of ink-flow and colour change. By selecting from a library of brushes, dips and papers, the stroke can take a wide verity of expressive textures.*

The brush has a one-dimensional footprint of bristles swept perpendicular to, and along a spline, defined by a number of nodes, each node containing position and brush pressure at that point. The pressure is linearly interpolated between nodes. Corners and turns in the stroke path are negotiated by rotating the bristles around the inner edge of the mark. Each stroke is laden with a specific amount of ink which is exhausted along the length of the stroke. This is renewed by 'dipping', also allowing the operator to alter the quality of the marks from stroke to stroke. The medium might not just be one shade, but a range of tones changing over the width of the footprint, thereby imitating the subtler nuances of Japanese 'calligraphic' painting [Figure 3]. Internal texturing within one mark can be further enhanced by texture-mapping (although not to good effect).

The spline is generated by defining points with a mouse or, preferentially, a pressure sensitive stylus. Animation is achieved by defining key frames and inbetweening.

### 3.4 Sasada's Hatching Techniques

Sasada's work [Sasada 87] is similar in spirit to Yessio's in its attempt at emulating graphical conventions seen in technical drawing and illustration. He states "the author would like to simulate hand-drawings using different computer techniques". He concentrates on the different types of cross-hatching seen in illustrations of landscapes, skies and seascapes. The motivation for the work comes from wanting to reduce the computational effort placed in rendering landscapes when making animations of cities in their geographical context.

*The development of all these techniques began with hand-drawings of natural scenery ... these where then analysed and ... appropriate algorithms were written to execute specific tasks.*

Most of the algorithms work by starting with a standard hidden-line rendering of a regular mesh



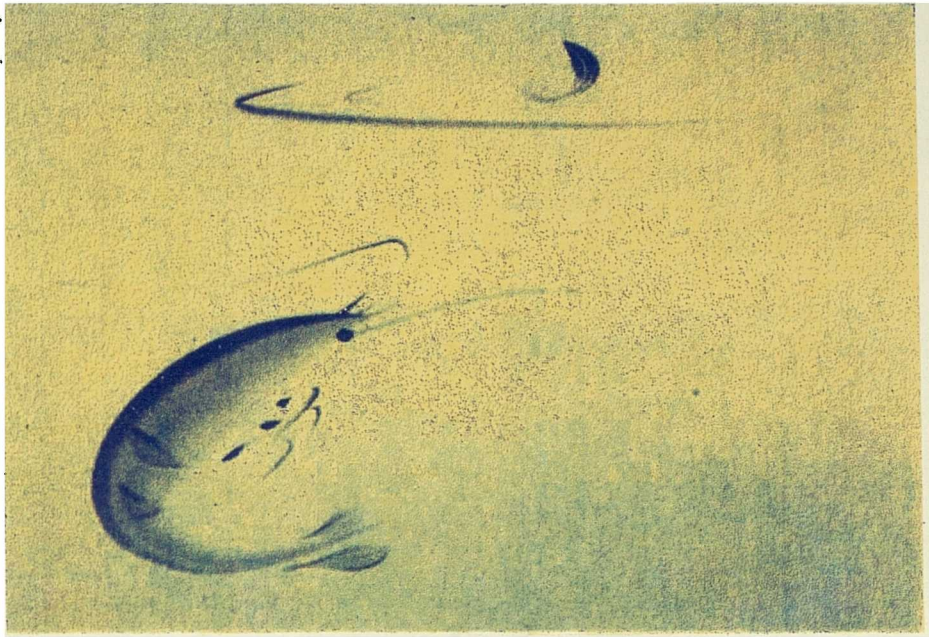


Fig. 3 From "Shrimp and Leaf" animation by Steve Strassman



Figure 6. Combination of ridge and waterflow line techniques

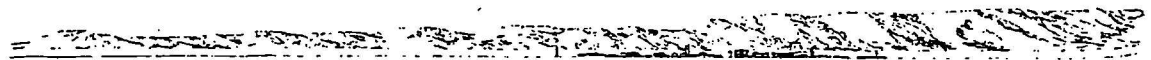


Figure 7. Combination of ridge, waterflow line, and shadow techniques

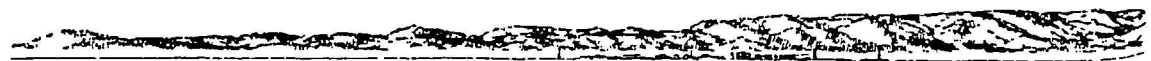


Figure 8. Combination of ridge and shadow techniques

Fig. 4 Landscapes using water-flow lines by Sasanda

which represents the mountains. This is then stochastically perturbed to create a natural hand-drawn look or by processing sharp changes in mesh direction to keep only ridges. A convincing hand-drawn technique emerges from calculating "water-flow" lines instead of a simple mesh. These are meshes of lines whose axis are defined as orthogonal to the mountains height contours, i.e. directions in which water would flow down a hill side. Different line thicknesses are used to emphasise light and shade [Figure 4].

For closer views of mountains and natural scenery Sasada uses "2.5-D" representations of trees and foliage to cover the landscape. These are 2-D vector representations of trees that are shifted, scaled, rotated to be facing the point of view and then drawn with varying line weights to represent forests. These techniques are not as convincing as the more distant mountain views because the repetition of more complex graphic elements is too apparent and animation may betray the fundamentally 2-D nature of the representation. To solve this he suggests using automatic stochastic tree pattern generation of the sort devised by Lindenmayer [68] or Aono and Kunji [84].

### 3.5 Pierre van Berkal's SIAS

The Strokes Interpreted Animation Sequences [van Berkal 89] system is designed to animate 3-D models composed of painterly brush strokes, the strokes themselves existing in 3-space.

The system not only generates marks, but also concentrates on the automatic motion control of shape to sounds. In the case of the video "Piece for 32 Non existing Voices" stroke-composed animated heads articulate with realistic facial movements in synchronisation to a synthetically produced voice. Great consideration is given to accurate correspondence between the phoneme and the facial expression. Following Strassman, van Berkal generates marks along a predefined spline, though in Berkal's case the modelling of the spline occurs in 3-space and is projected into 2-space for mark-rendering. Once mapped onto the 2-D spline, a polygon defining the shape of the whole mark is filled with parallel bands of colour which can experience changes over their own length as well as over the length of the stroke. Certain problems are encountered when the filling of the mark polygon is oriented in certain ways as the filling technique starts to split up to form highly visible bands.

### 3.6 Barnsley's Fractal Compression Images

Several images emerging from this work [Barnsley et al. 88] seem to suggest that the system can create convincing painterly renderings [Figure 5]. These are derived from digitised photographic

images. A picture of a Bolivian girl wearing a hat is particularly impressive. Barnsley's main concern however is in the high compression ratios offered by the algorithmic encoding using fractal, or more specifically Iterative Function System, methodologies. The long-term aim is the real-time compression and decompression of photographic imagery for broadcast, hence the visual distortions arising can be viewed as an unwanted artefacts.

Barnsley's research group has always been cautious of revealing its techniques. The compression technique relies on the mathematical concepts of collage theorem [Barnsley & Hurd 92], and the ability of the user to interactively help the algorithm make good choices.

*The collage algorithm ... provides a means for interactive geometric modelling using IFS (iterated function systems). In the two dimensional case, the input to the algorithm is a target image; for example a polygonal approximation to the boundary of a leaf, a digitised image of a leaf, or a representation of leaves on a branch. The output from the algorithm is an IFS code which, when input to the measure rendering algorithm produces a textured rendering of the original...*

The collage theorem guarantees that any IFS using these affine maps has an attractor that looks like the original image. It is assumed that the measure rendering algorithm is generated at the same time as the IFS .

Although it can be plainly understood how this technique would be very appropriate for conveniently self-similar images (leaves, shells etc.) the actual templates used for the girl's head cause heavy distortions in the end image because of the non-self-similar nature of the original image even though much of the encryption for the image has, we assume, been defined by hand. As Barnsley states

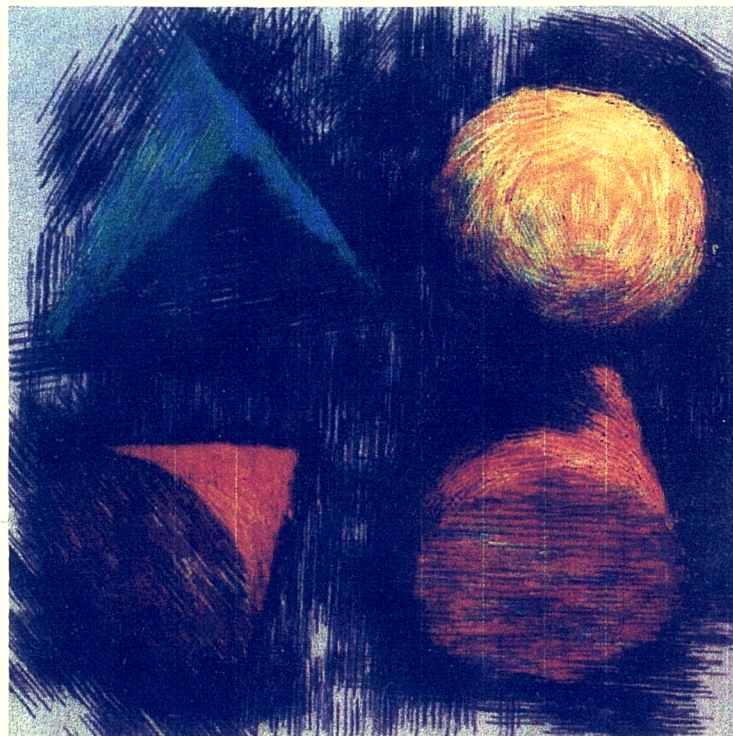
*"the closeness of this rendition to the desired image depends on the accuracy with which the user is able to solve interactively a certain geometric problem",*

just what this involves is unclear. Barnsley has announced that the interactive part of solving geometric problems can be automated – this would be necessary for the compression technique to be useful. While this betrays the limitations and frustrations of fractal compression and fractal rendering techniques it presents a novel method of abstraction that seems to re-symbolise the image by emphasising components as discontinuous within the whole.

### 3.7 Haeberli's Painting By Numbers



*Fig. 5 Bolivian Girl's head by Barnsley et al.*



*Fig. 6 Ray-traced and repainted objects by Paul Haeberli*

Paul Haeberli [90] describes several techniques for generating “painterly” renderings of digitised images and 3-D geometries. Firstly, frame-buffered images are re-painted using a range of marks, each created from a moderately simple graphic element, which samples corresponding points in the image buffer and recolour themselves accordingly. Using a range of different marks within the same image, divisionist painting styles are achieved that have a similar visual aspect to certain paintings. The marks are parametised thus:-

- Location - x, y on the image surface
- Colour - Red, Green, Blue, Alpha
- Size - Length and Width
- Orientation - the angle of the stroke
- Shape - the look of the brush stroke

A user interface is provided for real-time hand interaction with the system whereby the user can guide and direct strokes over the blank canvas and change their size and direction. The user has the opportunity to focus the activity of painting in several ways; in one implementation the depression of the mouse button causes randomly placed large marks to tighten up around the mouse position over time, as they do so the marks get smaller. An alternative to this is to have the tightness of focus of the marks and their size inversely proportional to the speed of the mouse movement, hence rapid movements of the mouse cause large widespread changes while slow mouse movements cause highly localised change. Further to this, several tools for automating the production of paintings are added. A brush stroke display list can be generated; which can be re-used to generate a sequence of visually consistent images from a sequence of frame-grabbed images. The automatic re-orientation of strokes over an image is implemented by establishing relationships between gradient of tone and brush direction. The tonal gradient can be calculated from the original image or from a separate underlying luminance map.

The interactive painterly rendering of 3-D geometries is achieved by replacing the simple digitised image of the above system with a screen-sized port into a ray-tracer. Colour and surface geometry of an object are calculated in real-time by firing rays into the scene at the mouse location rather than sampling pixel colour values. Using this method surface geometry and other material properties are available for various use. The geometry is used to automatically re-orient the direction and size of stroke depending on surface normal and distance from eye to surface. Marks can be made to visually accentuate object geometry by orienting themselves along lines of curvature [Figure 6].



A separate technique is presented for creating divisionist renderings through the use of Dirichlet domains (or Voronoi Regions) [Preparata & Shamos 85]. An iterative searching technique is applied to these areas adjusting them to fit and colour themselves to a frame buffered image. The initially randomly placed polygonal shapes colour themselves to the average of all the pixels in the corresponding patch in the source image. Over a sequence of iterations the shapes relax into best-fit configurations, trying to best represent the underlying colour by looking to reduce the standard deviation of the averaged colour to a minimum. They are constrained by the fact that the initial number of Dirichlet domains cannot change and that the total surface area of the patches must remain constant (i.e. they must cover the whole surface). One image produced in such a way may take several hours due to the heavy computational nature of the problem.

### 3.8 Fractal Design Painter and other painterly systems

Probably the first available explicitly painterly technique included within any paint system was Quantel's inclusion of a "chalk" effect on its paint brush menu [Quantel 83]. This was achieved by a simple look up table deciding whether or not certain pixel ought to be omitted when painting a mark.

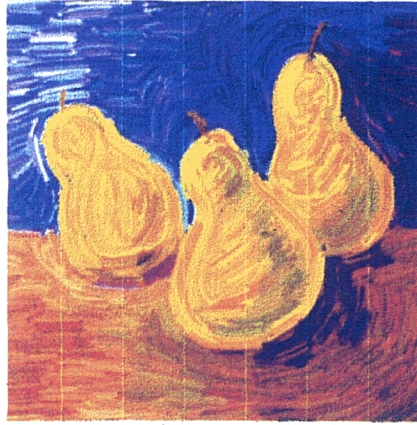
ImageWare's ImagePaint system also predates most well known painterly paint systems by several years [Image Paint 89]. This is an interactive post-processing package. A series of filters are applied to an image by sub-dividing it into many overlapping sub-images which are then treated by a filter and finally added to the output image with varying degrees of opacity. The subdivision of the original image is achieved through the definition of various meshes through which parts of the image are pulled. These filter meshes can be regular or random. These can be combined together with blanket effects such as edge-finding. By storing the meshes and filters as a set of parameters they can be applied to a sequence of images thereby facilitating some animation techniques, although for perceptual reasons these are not as effective as still images.

Pacific Data Image have developed an in house system for post processing PR images called "Render Stokes" to create painterly images [Figure 7c]. No details have been published although manual techniques for editing individual frames using various filters and brushes would produce the same result. It is assumed that part of the post-processing is done through non-interactive techniques.

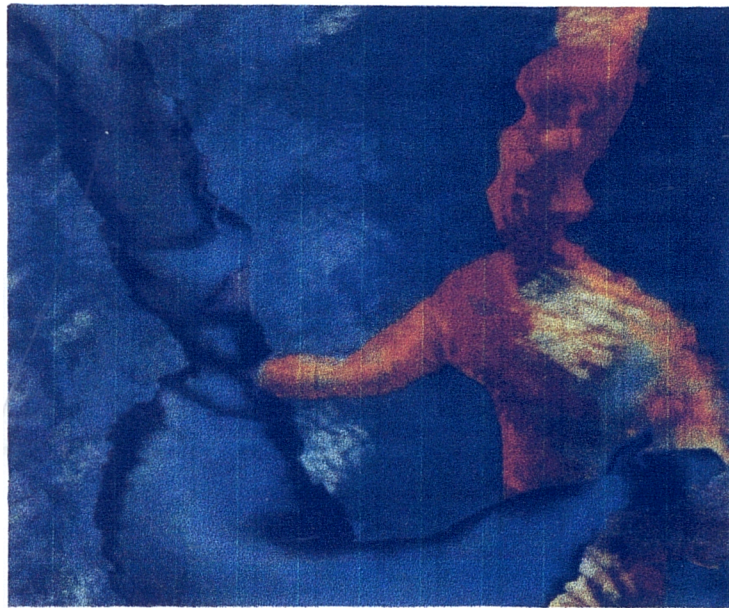
Certain image processing techniques for generating painterly images from digitised images are demonstrated by William Mitchell [92] using a basic image-processing package. These are a combination of one-to-one pixel mapping routines and random noise functions applied automatically



*Fig. 7a Post-processing for painterly ends using  
Adobe Photoshop with Aldus Gallery Effects filter*



*Fig. 7b Simulating Natural Media (via Fractal Design Painter)*



*Fig. 7c Post-processing photoreal computer graphics by Pacific Data Image*



across the whole image. Posterisation and edge-detection produce water-colour like images. Noise is added to the image through the use of random or wave noise functions [Figure 7a].

Fractal Design Painter [91] is an interactive 2-D painting system that specialises in “natural media” effects through emulating real-world media such as charcoal, ink, pencil, oil paint etc. It uses a range of painting tools, brushes, filters and textures. Many of these techniques employ types of randomness to “naturalise” the medium – for instance, small cycles through prefigured look up tables are used to generate the grittiness of charcoal. These are applied through standard interactive painting method although the user has the option to select large areas for blanket effects. By mixing these naturalistic effects in certain ways new image styles can be arrived at that bear little relationship to traditional techniques. There is some degree of automatic painting facility; the re-rendering of grabbed images (referred to as “cloning”) can be stored as a script. Several interesting and novel abilities are provided, such as the ability to push “wet” paint around on the surface with a “palette knife” [Figure 7b].

The degree of emphasis placed on the emulation of natural media – that the end product should look like a scanned oil painting, for instance – is indicated by the inclusion of surface embossing effects that emulate the positioning of lights at raking angles to the surface – thereby highlighting the tooth or weave of the canvas and impasto effects of oil paint. This is probably achieved by embossing through convolution filters then multiplying the luminance of the original image with the embossed image. Other more elaborate filters distort the image as if placing it under rippled glass. These effects are achieved by using spatially coherent 2-D noise generators to displace colours from their original positions.

Fractal Painter embodies most wholeheartedly an interactive paint package solution to many of the problems posed by the challenge of simulating natural media. There are several other packages which contain some of the same features within a more conventional paint-package framework that deserve mentioning. Ultra Paint [90] contains an excellent blending algorithm for layers of over-painting which gets very close to the look of ink or water colour on paper. Photoshop [88] has a set of plug-in filters, particularly Aldus Gallery Effects [92], which process selected regions of an image to create various pointillist and divisionist styles.

### 3.9 Cockshot’s Wet and Runny

This system [Cockshot 91] uses a cellular automaton approach to the modelling of a paint-like substance applied to a canvas substrate. This is extended [Cockshot et al 92] to consider how the cellular automaton structure might be employed to generate a bump map for creating a convincing 3-

D textured paint-like surface .

The initial paper presents a method for creating a paint-like substance which shares many of the characteristics of natural wet and sticky substances when applied to a canvas support. The substance bleeds, runs, drips and mixes; it dries-out and can change appearance as it does so, it can be affected by imposing "gravity", changing other environmental conditions such as temperature, and by changing the nature of the support. The implementation is broken into three main blocks :-

- The paint particle. This is an element of paint which represents a single unit of paint, its current wetness, colour and opacity.
- The Intelligent Canvas. This is a large grid of storage units arranged in a regular grid. Each unit can store a pre-set number of paint particles.
- The Painting engine. This is a process which visits every storage unit on the intelligent canvas and performs an operation on the paint particles it finds there while considering neighbouring storage units. It performs simple numerical calculations on each cell of the canvas and from the cumulative quantity of paint after redistribution determines the cell's colour. This is then written to the frame store.

A typical paint re-distribution process performed by the painting engine is to average quantities of paint with surrounding cells by spreading the load of stored paint particles as evenly as possible within a set time before the paint "dries". This causes the paint to bleed and mix. The movement of paint around the canvas is determined by sets of rules including paint drying speed and gravity directions. While the painting engine performs its task automatically and as quickly as possible, the original paint is intended to be applied via normal interactive techniques. Because of the heavy processing required for real-time dripping and bleeding parallel computing techniques are advocated.

The second paper presents a way of turning the quantity of paint at a particular cell in the canvas into a height extrusion of a regular mesh of triangles. Each triangle has a vertex placed over a cell and the vertices are Phong-shaded to the colour of paint present at that point. The height extruded mesh is then lit from a raking angle to create a bumpy surface. This would be interesting as an animation over time as the paint bled and dried.

### 3.10 Saito and Takahashi

Saito & Takahashi [90] presents several techniques for increasing the comprehensibility of 3-D rendered shapes. The premise is that most technical illustrations are clearer than photographic representations through their use of edge and hatching techniques. The final intention is to add

this extra layer of delineation on top of photo-realistic images.

The system renders a scene into a “geometry-buffer” (g-buffer) where in the 3-space co-ordinates are stored are an image-resolution raster. This can then be sampled on a per-pixel basis and image processing techniques applied to create various interpretations of the original scene. Sobel edge detection (2nd order differentiation) on the z field of each pixel produces good line drawings by finding discontinuities. Profile edges (silhouettes) and internal edges (corners) can be found separately and rendered using different filters (1st order differentiation). Generally, a good drawn effect is achieved by weighting the density of the pixel drawn to the level of discontinuity detected – hence, silhouettes are shown as strong lines whereas internal edges are finer. This image can then be used alone or re-applied to the photorealistic image to give it more clarity.

Hatching is achieved firstly through the development of a height-contouring algorithm. This is a kind of solid-texturing method which striates the scene at regular pixel intervals in world space using a thinning technique to constrain the density of lines in screen space [Figure 8].

### 3.11 Free Hand Plotting Group’s Wobbly Pen Plots

The simple technique of using a pen-plotter with loose, wobbly pens produces surprising and effective results [Bakergem & Obata 91]. A survey of what people think of these new “re-humanised” sketches has been conducted. After listing the many interpretations and opinions offered up in response to pairs of images – one a ‘normal’ pen plot, the other a wobbly pen version of the same image – conclusions were formed about the value of such a technique. The most surprising result of this experiment is that students, who previously fought shy of including any computer graphics in their portfolios, feel no inhibitions over including these wobbly plots. The paper states several possible explanations for the attraction of these wobbly images. One is that they are unique in that the deviations of the pen cannot be reproduced. Another is that they are simply more visually engaging – they seem to have more detail, or at least contain more to look at.

The images benefit from a synergy between the absolute control of the machine and the randomness of the wobble, the result being a supremely “confident” sketch [Figure 9]. As such they have apparently duped extremely experienced draftspeople into believing that they are very competent hand drawings, often to their later embarrassment when the actual truth is revealed. As a result of these investigations the group has produced a PostScript program for manipulating plot files which, in its completed form, provides more control over the effect.

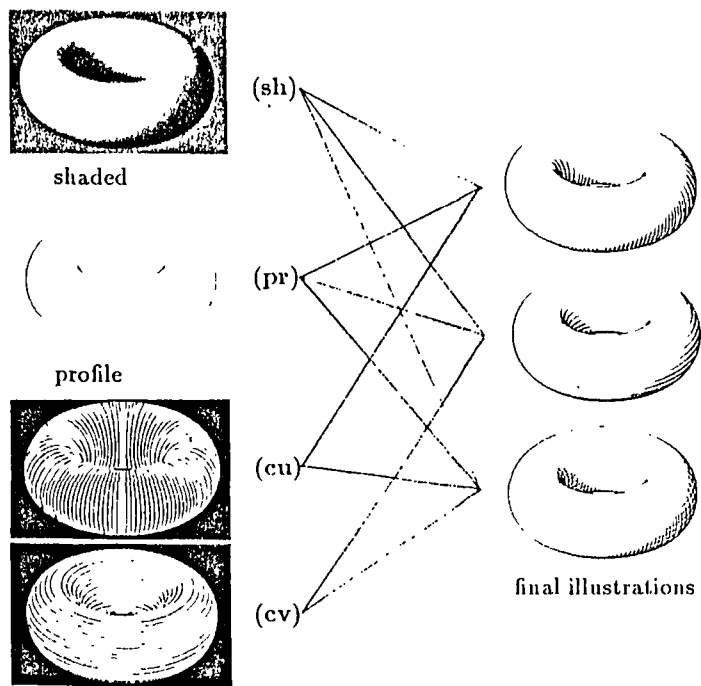


Fig. 8 Hatched and Edge-drawn donuts by Takahishi et al.

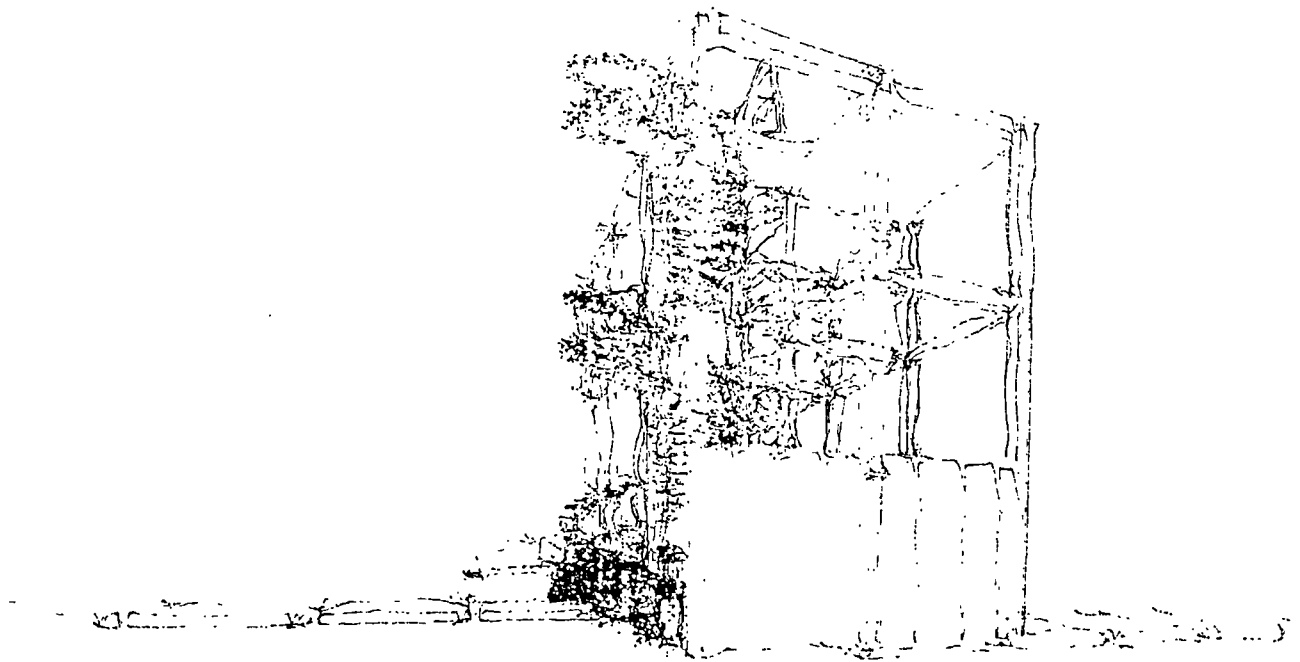


Fig. 9 A wobbly pen plot by Bakergem et al

*[The]approach I used is to actually modify the geometry of 3D model itself using random and fractal methods. Two characteristics of a human sketch are*

- 1. A lines endpoint doesn't meet exactly at an intersection with another line*
- 2. Lines are not exactly straight*

*... Both of the above effects are quite easy to simulate by computer. The first was done by extending both ends of line segments by a random distance between zero and a user chosen maximum distance. The wiggly line effect was generated by a well known technique from fractal geometry called "random midpoint displacement". The maximum displacement used to generate the wiggle is also chosen by the user.*

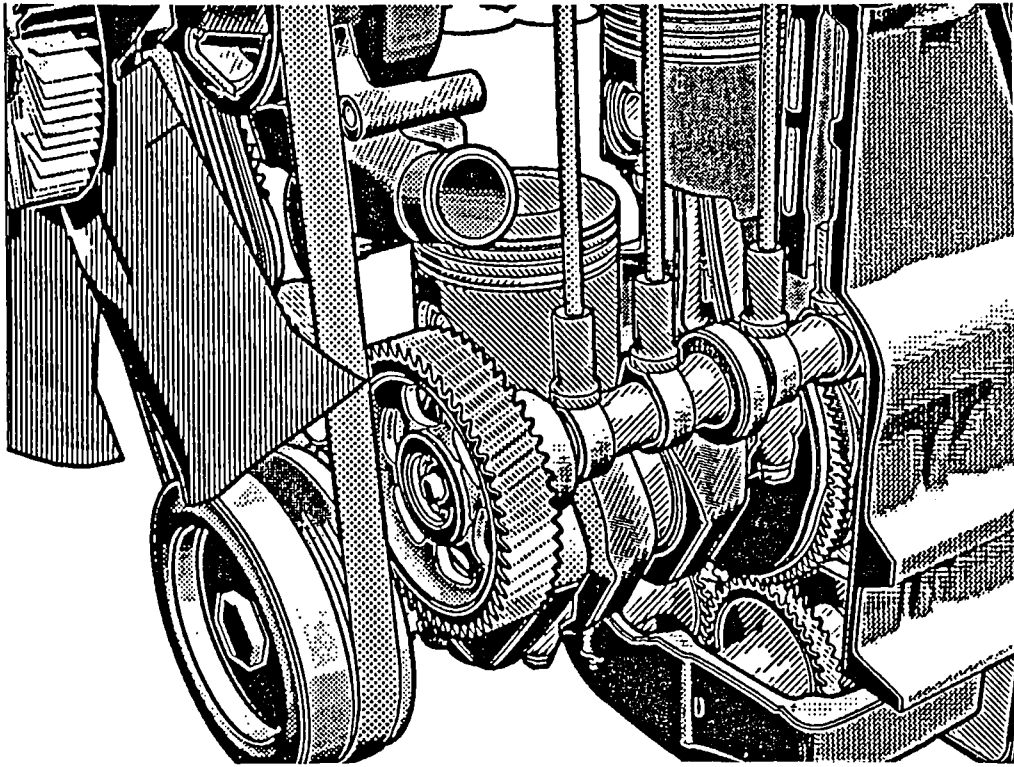
*Future work. Another noticeable effect of hand drawings is that the intensity is not constant along a line. Typically a line is darker at the beginning and lighter near the end. While it is not possible to maintain thin lines and introduce intensity on most printers, it should be possible to simulate this effect if a plotter is being used to create the drawing. Plotters that fully support HPGL can have the speed, acceleration, and even the force of the pen set for a line segment.*

[Barkergem 93]

### 3.12 5D's KATY renderer

This is an automated scan-line renderer which has the ability to produce technical illustrations using a range of techniques both photorealistic, graphic and painterly [KATI 93][Glazzard 93]. Many of these techniques are implementations of Saito and Takahashi [90] comprehensibility increasing algorithms although a simpler technique is employed for generating certain flatter styles of hatching. This technique calculates the luminance of a point in the output image using cosine shading and uses a mathematically defined screen-consistent hatch to render that point. The hatch is defined by intersecting sine-waves thresholded to two values – black and white. Altering the threshold alters the density of the hatch [Figure 10]

Added to this is the ability to automatically post-process images along the lines of Haeberli [90] and Fractal Design Painter [91].



*Fig. 10 Technical illustration styles applied to a model by 5D inc. using the KATI renderer*

## 3.14 Summary of Previous Systems

### The Morphology of NPR

This inventory gives some idea of the breadth and diversity of research and solutions appertaining to the many issues surrounding NPR. Despite this diversity a morphology of NPR is seemingly defined by only a relatively few distinct techniques.

- The inclusion of randomness or arbitrariness as a factor within the rendering process to create wobbles, distortions and noise of various sorts. [Mezei et al 74], [Haeberli 90], [Fractal Design 91], [Bakergem & Obata 91].
- The construction of images from indivisible pictorial subunits that are larger than the pixel, such as "brush marks" or polygons. [Strassman 85], [Mezei et al 74], [Haeberli 90]
- Certain techniques, such as edge detection and hatching that work on a per-pixel level to create artefacts within the scene which are not intended to be read as physical parts of the model, but part of the image, and which exist on a scale larger than the pixel. [Sasada 87], [Saito et al 90], [KATI 93]

The most important defining factor which characterises all these systems is that many of the algorithms used operate with regard to screen-space and object-space in new ways. Artefacts which are used to compose the image do not easily map back into the 3-D world (as with the projections used in PR graphics). Rather, they have a stronger relationship with the 2-D surface. The marks, image-processing and noise treatments of Fractal Design and Haeberli are applied to the image as 2-D effects, while the edge drawing and hatching of Sasada, Takahishi and the KATI renderer, while mapping more heavily to the 3-D model are constrained by screen-space algorithms relating to width and density of the marks in 2-D. An aspect common to all systems discussed so far then, is the continual re-examination, and further complication of the relationship between 3-D world space of the model and 2-D screen-space of the image.

### The Goals of NPR

The brevity of this morphology might suggest that the issues of NPR are confinable and easily comprehensible. This is not so; the cultural problems relating to worth and quality of these types of images – i.e. how "good" these images are, and how good they *might* be given the correct

circumstances – is highly problematic. In order to secure a critical position on this new type of imagery common assumptions, goals and achievements must be summarised, from which an argument positioning this type of imagery can be built, and from this (assuming already that it has problems) how it can be improved and matured into a valuable technique.

It can be seen from the appraisal of published systems that the default method in avoiding photo-realism is to create image that are in some way “like paintings” or “like drawings”. Mezei, Puzin and Conroy, Yessio, Sasada, Saito and Takahashi provide techniques that generate drafting or illustration styles similar to those seen in technical illustrations done by hand using line techniques such as cross-hatching and edge enhancement. The Wobbly Pen plot group likewise artificially manufacture images which are sometimes indistinguishable from real drawings. Strassman, van Berkal, Haerberli, Barnsley, Fractal Painter and Cockshott all generate images possessing certain qualities seen in painting. Sometimes this is explicitly stated as the goal of the system, as with Cockshott's and Strassman's deliberate simulations and Fractal Design Painter's post-processing. Other times it is achieved incidentally as with Barnsley's fractal techniques or Haerberli's methods.

The tendency to characterise non-photorealism generally as being solely constituted in painting and drawing as opposed to other forms completely new form of imagery – has its roots in the history of art. There is perceived a natural opposition between painting and photography [Gombrich 61], [Sontag 73]. The whole of Modernist painting can be seen, perhaps rather idiosyncratically, as having been profoundly re-motivated by photography in the sense that painting's previous project of mimesis (imitating nature) had been usurped. By way of retrenching, painters sought to emphasise those aspects of the medium that held out against photography. Various explorations into the physical properties of paint and the consequent admission of the pictorial surface suggest the seeds of abstraction.

*The development of photography was bound to push artists further on their way of exploration and experiment. There was no need for painting to perform the task which a mechanical device could perform better and more cheaply. ... (Traditionally) the painter was a man who could defeat the transitory nature of things ... Photography in the 19th Century was about to take over this function of pictorial art.*

[Gombrich 90]

Certain NPR images do indeed possess an initial likeness to various types of paintings and drawings. Sometimes this similarity is presented as the primary *raison d'être* of the system. For



example :-

*Fractal Design Painter ... simulates the tools and textures of natural media. With Painter, the eye actually sees the striated surface of oil paint, the saturated bleed of felt markers and the nubby richness of charcoal.*  
[Fractal Design 91]

However, the question as to whether or not images produced by NPR systems are like painting and drawing in any *significant* way, and can therefore claim the same significant advantages, remains unanswered.

This similarity to painting and drawing is enough to draw harsh critical fire, both aesthetically, formally and technically. And yet, because we do not fully understand what these images are really trying to achieve, they cannot be easily defended against such attacks.

Hence, it is crucial to the success of NPR as a field that it defines its objectives much more clearly, or redefines them entirely. This can only be done in light of its inherent advantages and limitations. As it stands, many systems suffer from what I believe to be a lack of critical perspective. They appear locked into a fatal trajectory through trying to convince that the artefacts produced are (or have been at some time) real paintings and drawings.

In the next section I would like to explore the problems with current NPR strategies more deeply. Two problems in particular need addressing. Firstly the problem that most current NPR techniques seem trivial and unable to articulate meaning, and secondly the problem that success is measured by similarity to painting and drawing, which will never be achieved. I will propose solutions to these problems.

## Chapter 4

# Criticisms of Current NPR Approaches

### 4.1 The Name "Non-photorealistic rendering"

As stated, the very name of the field is extremely unhelpful. It suggests that this new field is defined by an absence of something and may lead research to attempt a definition of the nature of the distinction between photo-realism and non-photorealism. This distinction is not at all clear and attempts to arrive at a definition have all foundered on those images which belong to both or neither camps. Painting as executed by Torentius, James Rosenquest, Salvador Dali or Gerhardt Richter embody properties of both painterliness and photorealism, as do the photographs (or photographically produced images) of Andy Warhol, Edward Steichen, Callum Colvin or Oscar Rejlander . As a consequence of this blurring, the intuitions of even a highly visually literate person are troubled when asked to sort out a collection of randomly selected images into two pile – photorealistic and non-photorealistic.

A clearer distinction can be made between a photograph and a painting – there are arguments to suggest that a photograph's relationship to the reality it represents is fundamentally different to that between reality and a painting. Sontag [77] touches upon a point I have not seen expanded elsewhere; that a photograph is a trace or print of reality, whereas a painting – even one meeting photographic standards of execution – can only be an interpretation. However, one of the points of this thesis is to discuss computer-based NPR as opposed to computer-based synthetic photorealistic images. Their respective relationships with the model (3-D, 2-D or digitised image) and, hence, the real world are the same so formal distinctions cannot be made on these terms alone.

I suggest the new name "Interpretive Rendering", as it points at least to a positive objective. It does not propagate troublesome arguments to the nature of photorealism. Though for the purposes of consistency and clarity I have chosen to use NPR throughout the thesis.

## 4.2 Why NPR is abhorrent to some

Because of NPR's initially perceived similarities to natural media, the degree of this similarity is often taken to be the measure of the NPR image's success or prowess. NPR is consequently often thought to be in "competition" with natural, hand drawn or painted imagery, or even proposed as a replacement for it.

But how can these images be compared to real paintings and drawings? They are limited to screen resolution; they have no physical body other than as a print-out; they only possess (as I will argue) the surface appearance of painting with none of the associated skill a painting embodies. From one perspective the problems of generating imagery that matches painting and drawing in a significant way seem so overbearing as to make the effort seem pointless. However, I will argue that from another perspective, if certain stratagems are adopted, significant improvements in the reception and positioning of these images can be secured.

## 4.3 Problem: Meaningless Effects Leading to Trivial Images

It is felt by many that certain digital image processing techniques give rise to "painterly" or non-photorealistic styles of imagery. William Mitchell devotes a chapter of *The Re configured Eye* to exploring image processing techniques, culminating in painterly styles [Mitchell 93]. Many commercial digital painting systems claim to be able to make images look like they have been painted and drawn, amongst them Photoshop, UltraPaint, ImagePaint and most concomitantly Fractal Painter. Typical painterly styles can be achieved by applying processes to the whole image in a blanket fashion. These are usually a combination of edge-detection to give the impression of drawn edges, posterisation to give the impression of block painting, blur and noise filters to give the impression of the irregularity of manual intervention.

Many of these systems now provide techniques for generated "pseudo-impressionism". This is usually achieved by point sampling a digitised image and recolouring a randomly shaped blob or brush mark accordingly. This mark is then placed at a corresponding point in the scene. The result is to break up the image into something like a divisionist painting style such as pointillism or collage. [Fractal Design 91], [Aldus Gallery Effects 92], [Monet 93], [Haeberli 91]. Haeberli extends this technique to include some amount of "intelligent" mark orientation and resizing.

The wholly user-defined application of painterly marks stands a chance of producing interesting and articulate images through the skill and judgement of the user, and this amounts to nothing more than using a traditional paint-box type system with some novel tools. In the case of interactive painting there is only a blank canvas and the imagination of the artist with no real automation. However NPR presupposes the presence of some extra source data to be re-interpreted (e.g. a buffered image or a 3-D model) which heavily influences the end-result through degrees of automation.

All of the systems reviewed provide some degree of automatic mark application. On the most trivial level they automatically give the mark a colour. More significantly, they may provide ways of scattering those marks around the scene without particular assistance from the user; Fractal Painter and Haeberli's Painting By Numbers provide scripted or automatic mark placement. The most significant levels of automation render entire scenes or parts of scenes without any user-interaction. In the case of the KATI system, the process is done on a completely automatic scan-line approach which fundamentally precludes interaction; parameters are set up by trial and error and a complete rendering executed.

The claim that these approaches can be used to generate imagery which is like painting or drawing in some significant way raises strong objections.

Many of the techniques discussed so far operate on the basis that painterliness can be obtained by a simple optical distortion of some previous photorealistically accurate representation. Unfortunately, the techniques and devices used in painting and drawing, accidental or not, are not distortions of some other, less distorted, form of representation. The types of elaborations and artefacts in actual painting and drawing are there for specific purposes and for particular historical reasons based on expression, style, abstraction, symbolisation and innovation.

This view that painterly styles are "distortions" has led to the assumption that the introduction of noise and randomness into an image fulfils many of the desired goals of NPR. A method for introducing photographic 'grain' to computer graphic images has been developed to "bring back the feeling into images" [Deeney 88]. Exploiting the limitations of half-toning techniques and hardware output devices provides another commonly used method. A highly successful architectural presentation made by Arup Associates for the redevelopment of Paternoster Square, London, included photorealistic renderings of proposed buildings that had been repeatedly dithered in order to soften and introduce grain to the images. [Hare 91]

It is unwise to assume that this sort of randomness , or noise, is equivalent to the surface complexities seen in painting and drawing. I would argue that the complexities and "inconstancies" seen in real painting and drawing arise from choice rather than randomness. When a painter wishes to represent something in a picture he chooses how to represent it, and while this choice may be arbitrary it is not random. Arbitrariness suggests a choice made from a number of acceptable alternatives, randomness suggests a lack of consideration. If one rendering instance differs from the next only in so much as there is an uncontrolled and unexpected number introduced to the calculation, then the resultant difference between results is literally meaningless. While the positioning and type of marks used in many paintings and drawings are perplexing and their causes perhaps unclear, we can hardly call them meaningless.

Using the techniques of Fractal Painter and Haeberli, then, whole areas of the image are subjugated to the same indiscriminate process. *Unlike the marks and gestures seen in real painting* they do not possess the same degree of *significance*. They do not embody any new level of information; they are completely blasé about the thing which they represent. One mark is no different from any other in its ability to articulate meaning. Underlying form or property is ignored in a pure regard for tonal patches of light, dark and colour. For this reason, even when this approach to image synthesis is capable of closely matching the visual aspect of some actual painting style – which of course is usually impressionism or pointillism – it completely fails in producing images which have anything like the subtlety or resonance of a painting executed by hand; the application of marks by the computer can never match the economy and deftness of expression of even the most amateur painter.

One might defend these techniques by claiming that the Impressionist movement employed a similar pure regard for optical sensation over that of underlying form. (Hence, these NPR images can at least be as good as Impressionist painting). But this reading of Impressionism is only partly true – “Treating the subject in terms of the tone and not of the subject .. rejecting the objectivity of realism, they had selected one element from reality – light – to interpret all of nature” [Rewald 80: 338 ] . It was never completely adhered to even by Monet, the movement's main exponent of *plein air* painting , who worked up many of his paintings from sketches in the studio, using the vast range of skills and techniques derived from historical precedents for emphasis and composition.

*Alluding to the these problems, Renoir used to say “Out of doors one is always cheating”. Yet their cheating merely consisted in making a choice among the multitude of aspects which nature offered in order to translate the miracles of light into a language of pigment and two dimensions, and also to render the chosen the chosen aspect with the colour and the execution that came closest to their impression.*  
[op cit. : 340-341]

There is little evidence to date suggesting that the image processing techniques of the type used by Haeberli are able to match the sophistication of process evident in any painting by hand. Even with an ability to orient, scale and recolour elements within the scene according to rules based on colour alone, this does not produce marks as articulate as those seen in real paintings.

#### 4.4 Generating more significant imagery through improved technique

In short, the measure of success of a non-photorealistic rendering system must partly lie in the degree of signification achieved between the thing represented and the way in which it is represented. This must be increased before a system can claim to be a *significant* interpretative representational technique. The methods offered by a system like Haeberli's provide only the thinnest icing over the top of some original image – they add almost nothing. A more synergistic reworking of the representational schema is needed, so that the thing represented and the mode of representation unify in a powerful and sympathetic way. This synergy is present in all good drawing. The technical nuances of modelling form, indicating surface texture and evoking subjective/poetic associations within the things drawn or delineated are extremely desirable abilities for a system.

Of course, most image processing systems are at an immediate disadvantage; they usually have no access to data relating to the underlying form and material properties of the scene. Automated photogrammetry [Beyer & Streilein 91] and "unlighting" techniques [Durisch & Anderheggen 1991] may extend the ability of conventional photography to provide techniques for determining more than the colour or luminance of a scene. Photogrammetry uses stereo pairs of images to calculate the depth of any point within the scene, while "unlighting" techniques recalculate the unlit optical properties of materials within a scene – these can be then "relit" from another angle, or used to determine material similarities between different elements within a scene. But these techniques are still highly experimental and their eventual solution far from certain. However it is a trivial matter, at least conceptually, to generate fuller descriptions of scenes from 3-D geometric models, where in the whole data set is available to the system, and editable by the user.

Access to geometric and material property information would be a significant improvement over current colour-only based systems. Considering the production of real paintings and drawings, for an

instant, artists have sophisticated prior knowledge about the world they seek to represent. They usually have a fundamental grasp of the geometry of things; even if something cannot be seen clearly, such as the distant trunk of a tree in poor light, the artist knows that it is roughly cylindrical and can use this knowledge to add descriptive detail to the rendering such as hatching and texturing. The artist also knows where the tree ends and the grass begins and can emphasise and clarify the distinction between parts of the scene by edge enhancement and using different techniques for different elements (e.g. leaves are drawn with clearly different marks to grass). Image processing and robot-vision image recognition techniques have a long way to go before they can be used to add this sort of knowledge to a representation reliably.

We understand then, that the artists adds information to the drawing which relates to aspects of the world over and above the purely optical. In order to deduce what the many devices and techniques might be that could be added to the capabilities of an NPR system – it is inevitable that an inspection of real painting and drawing be conducted and some form of analysis applied as this seems to be our primary source of present examples.

#### 4.5 Traditional Techniques for creating Significance: An analysis of Drawing

The types of representations chosen as a starting point here are those "realist", documentary styles of drawn representation which became the basis, later, for drawn architectural representational techniques. The etchings of Gustave Doré, Giambatista Giovanni Piranesi [Wilton-Ely 78], the school of John Soan, the school of Thomas Bewick [Cirker 62] are all examples of this oeuvre of representation and provide tough enough aspirations for any system development.

Many attempts have been made to analyse painting and drawing in formal terms. However, this has proven to be a very difficult task and there are several pitfalls. Modernist attempts often became irreconcilably embroiled in murky aesthetics discourses including those of the early formalists of the Bauhaus [Kandinsky 47] [Klee 40], and Maholy Nagy [Nagy 65]. More recently post-structural, linguistic and cognitive science tools have been applied to build models of artistic/creative production from which other works of a similar nature might be produced or understood [Stiney and Mitchell 78][J & R Kirsch 88]. However these projects have proven too limited to be used as general models for the larger issues of design on which they, perhaps, initially sought to shed light. In hindsight, clear prescriptions have not emerged for general design methodologies [Richens 94].

The approach adopted, in the light of past endeavours consequently avoids the general theory-based views of drawing production and considers the specific, example-based, study of image production. It considers images made by drawing and painting to be an accretion of a vast range of formulaic techniques. This approach does not do full justice to exemplary works of art; but it is in deference of the potential pitfalls of adopting a more sophisticated analysis of artistic production that this reductivist approach has been chosen.

In defence of this approach, it can be seen that even in the most elaborate representations certain definite schema are at work – the application of a definite technique for a definite result is witnessed. This may sound obvious, but the methodical systematisation of representational devices in painting and drawing is not often alluded to in academic studies of artistic production. Such reductions are generally thought to run against the grain of appreciating the work in full. These types of observation are often perceived as over fussy or damaging to the reputation of the artist by considering art works to be mere technical exercises.

The commonly held model of the artist's skill and method, and the one perhaps unintentionally projected by most Art Historical rhetoric, is what I call the “virtuoso model” of artistic production. The Virtuoso Model assumes that the artist accrues and sublimates an arcane skill, composed of some huge cache of unfathomable heuristic techniques which are driven and inspired by obscure intuitions. These virtuoso skills cannot be related or understood except through lengthy and immersive exposure, so that we get a “feel” for the artist's work; this understanding then acts as a backdrop to all further analysis.

This kind of mythology belies, like all myth, a good deal of truth; there is indeed embodied in the works of the artist Piranesi or Rembrandt a vast amount of talent and vision. But it is also true that draftsmen such as Piranesi or painters like Rembrandt collected a catalogue of standardised methods of representation which were applied quite coldly, without necessarily the effort of the virtuoso, in quite specific and predictable contexts. This collection of techniques would have been formed throughout a long and tedious apprenticeship, which would have consisted of many arduous re-renderings of formal studies in the styles of exemplary fore-runners. This method of schooling is today usually viewed as over-constrictive. Our reticence to analyse images on this atomic level may stem from a fear of being labelled myopic or detail-obsessed. On this very issue of a seeming reluctance to acknowledge the merit of formularisation within representation Gombrich states, (beginning with a quote from an study of drawing by F. C. Ayer [Ayer 16] )



*"The trained drawer acquires a mass of schemata by which he can produce a schema of an animal, a flower or house quickly upon paper." The whole temperament of art in our time revolts against such procedure. Have we not just struggled free from the dreary and melancholy methods by which Victorian boys were taught to draw... The historian knows that such revulsion from the formula is a comparatively recent development.*

Gombrich's essay "Formulae and Experience" [Gombrich 60a] generally concentrates on the more macro-scopic aspects of image making (how to draw a head, a lion and so on) and holds a contempt for the formulaic that can only be correct, calling it "the pathology of portrayal" which implies that it is akin to some kind of illness. None the less he admits that all representational schema ultimately depend on formularisation.

*Not that we need doubt that all these artists really strove to forget formula. But the sober observer will realise there is all the difference between trying to forget something and never having known it*

i.e. that formularisation is inevitable and can only be dealt with by degree and then, probably, only by replacing one set of formula with another.

In our latter day revulsion of the formularised method within representation we may have fostered an over-romantic view of what a drawing or painting might be made up from, particularly in the context of *mimesis*. While the overall vision for a particular piece and its resultant meaning-effect arise from a deeply human tangle of talents and influences, the actual specifics of representation at a certain low-enough level become much more deductible and their analysis more tolerable. While we can legitimately worry about the formularisation of drawing heads, flowers, clouds, birds, as soon as we go one stage lower in the scale of representational schema these anxieties start to evaporate. The same cross-hatch in a Piranesi drawing occurs in Doré as in Morandi as in any number of artists' work right up to the present day, and we do not feel the constraints of an over formularised technique at hand. At this microscopic level we are prepared to allow for universal formularisation.

It is one of the conjectures of this thesis that, at an atomic level of image production, there is much going on that could be given assistance through algorithmic techniques. The more macroscopic levels of general forms within an image and the general composition of an image overall are not the concerns here because they ought to be dealt with through other methods - abstraction of form is perhaps to be a modelling problem and cannot solely dependent on the method of rendering; composition ought to be a problem for the user and not the machine.

This atomic view is supported in part by the existence of all those drawing exercise books,

techniques books and technical drawing apparatus such as Letraset, Instantex and complex hatching apparatus which aim at prescribing good and clear methods for representation, particularly in design work [Gill 76 ]. Several books aimed at architects provide a large ranges of rendering techniques that could be regarded as the first potential goal for a non-photorealistic rendering system [Cusack 1890] [Porter and Goodman91 ] see [Figures 11a and 11b].

Examples taken from standard methods prescribed in Drafting Techniques manuals only indicate a small sample from a wealth of techniques that could be one of the initial goals for a non-photorealistic rendering system. To create a taxonomy of these techniques would be an enormous task and one that is outside the scope of this thesis. However it may be a very valuable exercise in the long-run and one which would not be without end. Perhaps the range of techniques used in the micro-structure of image making is smaller than one may initially expect. For example, the rendering methods used in etching and pictorial drawing are often quite similar; tone has to be forced from various hatching and half-toning techniques. These are simultaneously used to delineate form and surface texture. As argued, the differences between a Doré and Dürer mainly lie in the larger styles of representation – in which they are very different. But in the micro-structure of the image extremely similar techniques are used, even though one is a drawing cut into copper-plate and the other a drawing cut into wood.

Techniques for shading, delineating form and describing surface texture are all reasonable areas for study and acquisition by a NPR and methods for their integration ought to be considered as reasonably central to the overall system design.

## 4.6 Image Space vs. World Space

Something which emerges when studying these types of drawn configurations is the fact that they can be considered as being effects which occur completely in 2-D rather than world space, they do not strongly relate back to the 3-D world. Drawing techniques are as much concerned with pen thickness, line frequency and subtle mannerisms that occur on the image surface, as they are with accurate perspective.

This awkward interplay between 2-D and 3-D can be witnessed in all drawing. I will use two examples. The first is an engraving by Gustave Doré, from “London”, in which the endless wall connecting a series of terrace houses sweeps into the distance presenting us with many levels of continually shifting scale and orientation [Figures 12 a,b,c]. Onto this surface Doré has drawn bricks using a roughly consistently sized pen. The actual marks in the foreground are the same size as those in the background, they have roughly the same width, colour, chalkiness and shape – it is

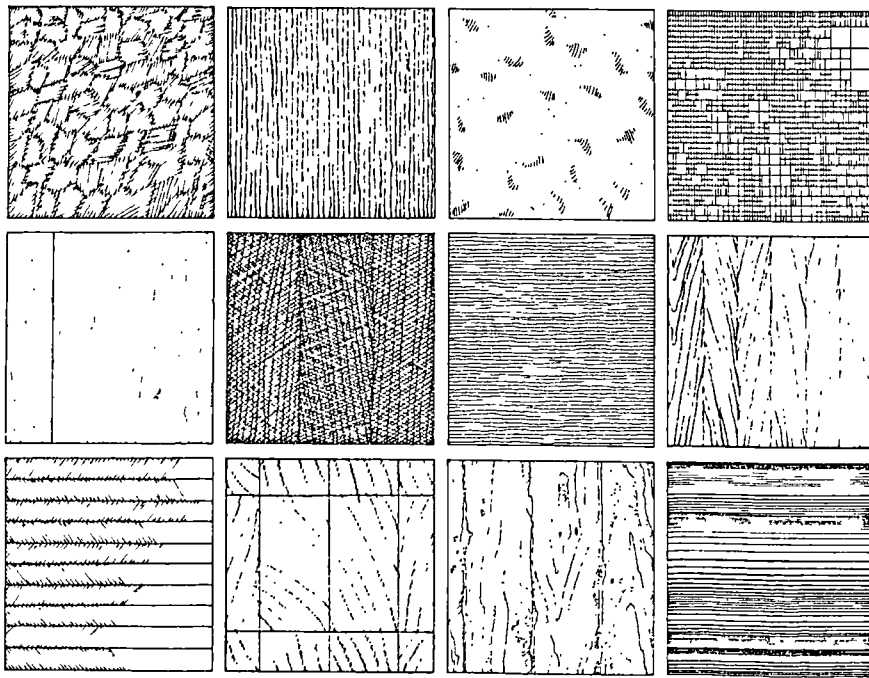


Fig. 11a  
Standard representational devices used in design work from "Design Drawing  
Techniques for Architects, Graphic Designers and Artists" Tom Porter and Sue  
Goodman, Butterworth Architecture, 1991 Oxford UK

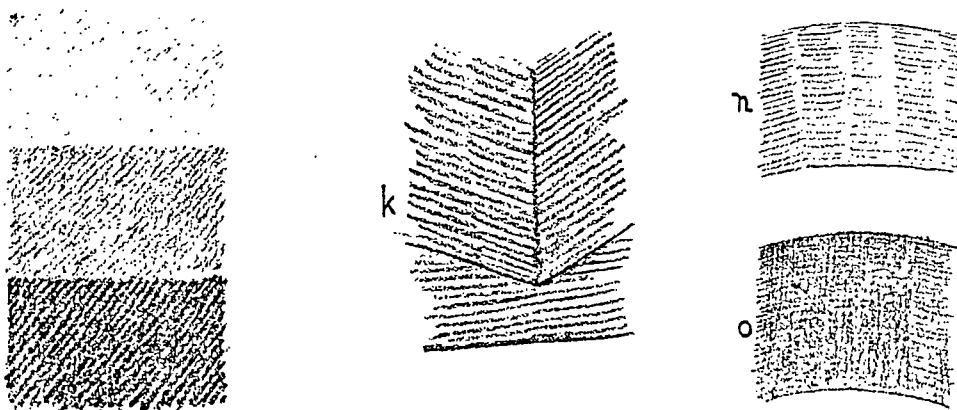


Fig. 11b Taught techniques from a Victorian primer (Kusak's Shading)

their configuration which has changed. On the whole it has changed to ensure that the density of marks in the foreground is roughly the same as those in the background – hatching is not scaled – it remains consistent, perhaps with a little thinning to suggest atmospheric fading – while the converging brick courses need to be thinned out as they recede to avoid overcrowding. The effect is to constrain the density of the marks in screen space while being authentic to the perspectival facts that bricks get smaller as they go into the distance and change their orientation and projection as they curve round a surface. 2-D and 3-D considerations and demands intermingle and battle continually. In fact the problem is even more complex than this; we can see quite clearly that the representational schema used for bricks in the foreground does not smoothly interpolate according to some single parameter set as the bricks recede into the distance. At certain points in the drawing discontinuities occur that mean the bricks are drawn in a new and completely different way. This can be seen by comparing the nearest bricks to the furthest – they are nothing like each other. Yet the overall effect to our eye is quite convincingly continuous suggesting that the discontinuities have been blended subtly into the image.

The second image is Piranesi's vedute of the S. Maria Maggiore [Figure 13 a]. The ceiling mouldings, as well as other features, recede to such a degree that the ones in the very foreground have around sixteen times the image space as those in the very background. The same geometry is represented using the same tool, but in a sixteenth of the area. Figures 13b and 13c have been rescaled so that the mouldings appear to occupy the same paper area for the sake of comparison – the necessary modifications for rendering at their actual different surface areas using the same tools becomes very apparent.

This indicates that some effects are medium-specific (completely disregarding the 3-D aspect) such as the engraving tool size, while others are world-space consistent, continually changing with regard to the 3-D information, such as the general direction of visible brick courses which form lines towards the vanishing point. And some effects lie between the two, like the screen-space thinning of the brick courses as they recede. Other effects are not smooth, and treat the 3-D data with something less than total regard. This presents a very complex set of interplaying features, however I feel that with time these problems can all be overcome and a general framework for generating images of this kind will emerge from a series of experiments.

This conflation of 3-D projection and 2-D technique relates back to the crucial distinction between the paradigms of PR and NPR. Photorealistic Rendering considers all its visual aspects to arise from projections of 3-D space, whereas Non Photorealistic Rendering has a much greater regard for things happening on the image surface; the 2-D. This suggests that a system aiming at the sorts or

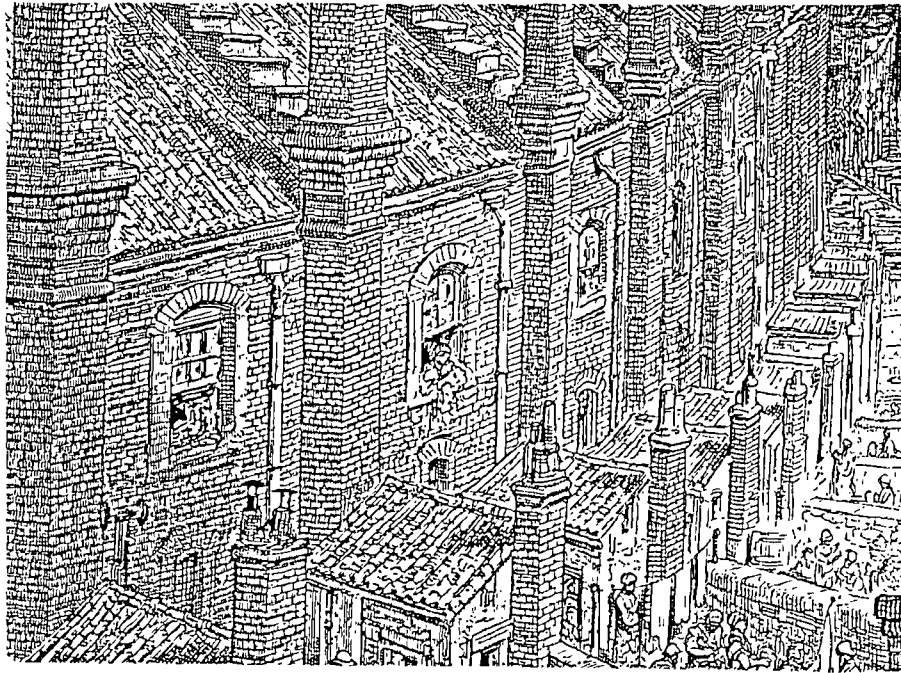


Fig 12a From "London" by Gustave Doré

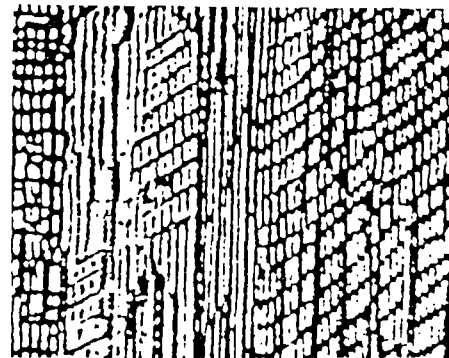


Fig 12b & 12c Details: near and far brick work

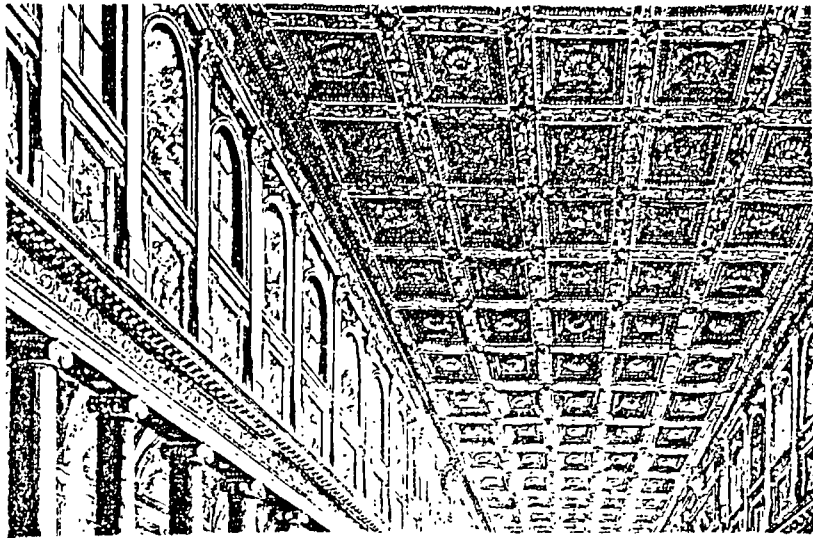


Fig. 13a Piranesi's *verdute* of S. Maria Maggiore, Interior

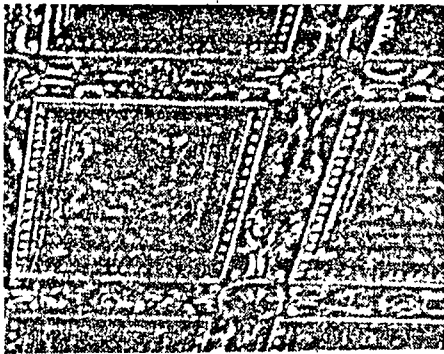


Fig . 13b and 13c  
From the Foreground and From the background rescaled to be roughly the same size

representation seen in drawing would spend a lot more time on screen space operations than previous renderers. This approach is adopted by several previously discussed systems wherein the final screen-writing operations consider the image to be a 2-D surface, which is being "painted", on a larger basis than the pixel [Haeberli]. Many of the operations will undoubtedly consider large patches of the 2-D image performing relatively complex operations that embody some representational complexity based on user defined descriptions of certain techniques. This makes non-photorealistic rendering actually more problematic than that of PR because there are general solutions used for rendering 3-D scenes that prove to be robust through their adherence to 3-D geometry and a relatively confined part of the process projects the geometry into 2-D after which it can be largely forgotten about. However the problem is aggravated in non-photorealistic rendering through the continual and rather badly defined conflation between 3-D operations like scaling, projection and perspective and the screen-based 2-D operation such as drawing configurations of marks and treating the image as so many patches of light and dark.

One emergent area of concern for NPR development is to clarify this conflation between 2-D and 3-D. Methods concerned with accurate reportage of the projected scene and methods concerned with how those projections are interpreted to "sit" on the 2-D surface must be clearly defined and combined in flexible and complex ways.

## 4.7 Drawing as a Hierarchical Process

Some of the techniques observed in real painting and drawing seem quite simple to describe at least, such a pointillism or block colour techniques. These techniques are quite homogeneous and invariant from one instance to the next and rely on the application of one level of relatively simple configuration. Some techniques, however, tend to be more complex and "nested", such as the techniques for brick or stone work. A brick, for instance, will have several levels of internal detail such as the brick outline, internal shading and cross-hatching. There might be techniques for drawing many bricks quickly to form, what looks like, a wall using, again, a hierarchy of representational devices. To outline this hierarchy of structure would give us an insight into the sorts of configuration we may want to be able to define for an NPR system.

At the bottom level of detail is the single mark. This can be considered the lowest graphical subunit for our purposes. Naturally the mark is a very complex thing in itself, but it is felt that it can be broadly parameterised and following Strassman and Haeberli this has been shown to be the case for most small marks such a pen, brush and engraving marks. Larger types of marks – those things which are still indivisible – but which have more complex internal structure due to variances over the area covered have not been fully investigated although image processing

techniques of raster patches have shown promise. An approach to these types of marks which are found in ink and water colour washes would be desirable.

Let us suppose we have a general interface for creating a mark of any size, colour, orientation and characteristic. This could be supported by controlled processes which offer variations round an ideal in a way which emulates the complexities and discontinuities seen between one mark and another in many drawings.

The next problem, and this is the largest problem to be solved, is to create some sort of framework for defining how the marks are applied over the surface. We have noticed some specific instances of how marks are used in the example of the receding bricks of the Doré engraving. Some phenomena have been observed within the representation of a single element i.e. the wall:

- Complex configurations built from simpler sub-configurations in the instance of drawing a single brick – outlines and cross-hatch may be used.
- These are configured in relation to one another i.e. the half-step tessellation of brick coursing under the guide of perspectival effects
- These configurations may interpolate some of their aspects as they “move” over a surface, such as the general shape of the outline and the perspective effects generated through cross-hatching, while others do not change like the overall size of the engraving tool.
- Sometimes completely new configurations are required to represent essentially the same thing at different scales and orientations.
- Subtle variation is seen between one mark and another, and one brick and another that could be attributed to human inconsistency or a wish for greater image complexity

The rendering of brick walls, or ceiling mouldings is actually quite a complex piece of drawing to attempt via completely automatic NPR methods. However these are just the sort of laboriously executed drawings that an NPR ought to be aiming at. If these specific examples could be resolved then obviously less complex types of drawing could also be done. But it may also be true to say that the problems posed by the brick wall are those posed by many seemingly more complex drawing problems. A crowd scene of great detail, for instance, may be achievable using similar techniques to that of the brick wall, perhaps with the inclusion of some z-buffer techniques to put one person correctly behind or in front of another. This also poses the fascinating problem of defining a characteristic person in some way, which of course seems harder than defining a characteristic brick, for the purposes of this kind of rendering. An apparent solution to the overall problem may require the explicit definition of at least one figure and methods for creating variations thereof. It



would appear that the repeated use of similar or even identical figures and motifs within the same image to generate complex and visually rich results is an effective rendering technique [Figure 1][Figure 33]. Foliage, plants in general, geological formations, urban texture and cloud formations all seem temptingly possible.

## 4.8 Summary of Techniques for Improving Image Quality

Several techniques can be implemented to improve the degree of representational articulation of NPR imagery. Techniques that ought to be initially provided are:-

- Methods for generating single marks of different kinds in a very general way
- Methods for explicitly defining layered configurations of these marks that respond to perspective, tone, colour, material type and surface texture in parametised ways, or swapping in completely new drawing-techniques if required
- Methods for defining stochastic variance in described configurations
- Methods for attributing specific configurations to specific parts of the scene
- Methods for creating the correct relationships between adjacent or over-layered configurations . These techniques include clipping against material boundaries or previously executed drawing. More complex forms of clipping will use depth sorting to achieve correct layering

A large area of concern is the articulation of surface geometry and material type. In order to do this, the data available to the system must be generally increased above the optical. The information that could partly match that available to the human artist counterpart is the inclusion of geometric information and the distinguishing of one object or material from another.

The introduction of these techniques would greatly improve the information content of the image and therefore its level of articulation. However these techniques alone will not answer all the problems relating to image quality. Certain problems, relating to how we look at NPR images, are more culturally deep seated. NPR images are often compared with real paintings and drawings for

the final judgement. I have heard many people make comments like "it's good – but its obviously done on a computer", or "it's not as good as a *real* painting", as if the likeness to painting and drawing is the sole yardstick of NPR's success. In the next section I argue that it is the comparison between NPR and painting which is at fault, rather than NPR itself.

## Chapter 5

### Mature NPR is not Painting Simulation

#### 5.1 Problem 2:

##### The comparison of NPR to real painting and drawing

It can be seen that most NPR system produce images that look in some way as if they were drawn or painted by human hand with real (physically speaking) materials. As such they start to look like reproductions of real paintings and drawings. Whether this visual similarity to real painting and drawing is a side product of some, as yet, ill-defined process or is the ultimate goal is an important question. If, on one hand a system is built which intends to produce images which are meant to be visually *identical* to paintings and drawings then it is inevitably doomed to failure; it cannot be done. On the other hand, if a system produces images which are *incidentally* like paintings and drawings, then one is inclined to see this incidental similarity as the side-product of some primary strategy. If we can deduce the true goals of this primary strategy then we will better understand the true objectives of NPR.

This visual similarity to painting and drawing has been historically problematic. It has given rise to the accusation that NPR is attempting to be "painting simulation" and it is assumed by many to be therefore attempting a surrogate for natural media and hand drawn or painted imagery.

If painting simulation is the ultimate goal of a system it is consequently obliged to convince the viewer that the artefact observed is in fact a real painting or drawing, or has been a real painting or drawing at some time, even if now it is seen through reproduction. As such, the makers of these images must expect them to be regarded as real painting and drawing and have them judged accordingly.

With respect to several of the reviewed systems it is clear that this visual similarity is not a side product of some other process but an intended property. We can point to several examples of explicit painting simulation, where the image is intended to deceive the viewer into believing it is, or is a

reproduction of, a real painting or drawing done by human hand using real chemicals.

Strassman's **Hairy Brushes** [Strassman 85] system aims at a very accurate simulation of how brush and paint acts on a surface. It makes for a good but typical computer science project in its proposition that by simulating elements thoroughly, the combination of elements working together produces a complex but controllable result. Paper, bristles, ink and hand-motion are simulated – out of which happens to fall a 'brush mark'.

While Strassman's work ultimately seeks validation in the paradigm of computer science it has predicated many subsequent designer/artist oriented systems and is often cited by other people such as Haeberli [90], van Berkal [89] and Cockshott [91][92], as a valid direction for further development. As such Strassman's approach to simulation has underpinned much NPR development and research.

Fractal Design Painter also goes to excessive lengths to simulate the specifics of natural media listing a range of "authentic" techniques such as charcoal, crayon, felt pen, oil paint etc. Added to this are a range of surfaces which aim at simulating the grain of various natural media supports – soft, open textured cotton, rough canvas, water-colour paper achieved through a sort of embossing that gives the impression of a surface under a raking light.

A phenomenological objection arises from the very act of simulation which may constitute an affront and give rise to accusations of deception. The *purported* body of labour and the *apparent* actions of wet ink on paper evident in the image are both, fundamentally, deceptions. Sumie painting particularly is a very human-centred, even performance-based, art form[Sze 77] . Sumie emphasises, to a high degree, the mythic act of artistic creation which Harold Rosenberg [69] and Brandon Taylor [87] defend as the source of value in many works of art . Through seeing the marks and re-discovering the performance that created them we retrace the momentary, transcendent state of the artist. Another crucial source of our engagement with painting and drawing comes from the sensual response to the physical materials – their complexity, subtlety and process.

But – using Strassman's images as an example – there is no artist to "relate to", only a machine. Nor is there any ink or paper to respond to, only a grid of pixels. Despite this, Strassman's images seek to impress us as *real* Sumie paintings. Strassman's work, Fractal Painter and the Wobbly Pen Plot Group (although they, at least, acknowledge this deceitful aspect), seem to imply that images can be entered into a sort of Pictorial Turing Test [Boden 90], where by if they match the real world counterpart in their apparent aspects (surface appearance perhaps via reproduction) then they

may as well *be* their real world counterparts. This poses many problems to those who associate the value of an image with the skill and judgement which went into its production. This is, after all, not such an obscure system for judging the merit of an image. As a consequence of this attempted deception the simulation of paintings starts to take on the same complicit, self-knowing tones evident in other simulations such as Weisenbaum's ELISA [Weisenbaum 76]. Any amount of simulation, no matter how convincing, will always beg the question "why do it?" – is it because, as Baudrillard suggest of AI – that we despair of the real? [Baudrillard 93]. While we may despair of the burden of our own intelligence and so wish to alleviate ourselves of certain intellectual tasks, I do not believe the same can be said of painting. Not only this, but the simulation of painting may well prove to be as elusive and misguided as the simulation of intelligence [Penrose 92]; certain things, the aspects of which are already formalised yield to simulation in useful and informative ways; the problems decompose well and theories can be tested against the model. Painting, however, with its unknowable complexities, intimate reliance on the performance of the artist and the sumptuousness of physical materials does not decompose well and so is intrinsically unsuited to simulation.

The problems of simulation are further illustrated by Cockshott's **Wet and Sticky** [92] system. Quoting from Cockshott: The system arises "from an expressed dissatisfaction with the sterile quality which tends to pervade images produced with electronic paint systems". In comparison to real paintings, electronic paintings display "a much narrower range, a lesser variety, and reduced level of complexity". **Wet and Sticky** will produce images "whose marks possess the same degree of complexity and variety as those found in real paintings" because the system closely simulates the chemical behaviour of paint on a surface. As a consequence he claims his system's marks are better than those exhibited by Fractal Painter etc. because they match the complexity of real paint marks. In one important way this is not true; the complexity of real paint marks lies in their physicality – the fact they possess the resolution of the chemical atom and their physical depth, that they are not films but almost sculptural in their depth. No matter how Cockshot seeks to produce his digital mark it will always have the resolution of a digital system, and will always be a film one colour deep. The extra complexity in Cockshott's images is not to be found in the isolated final result but in the process of image generation.

What Cockshot is offering here (although he does not state it clearly) is not only a simulation of a completed painting, but a simulation of the dynamics of applying paint to a surface (He goes so far as to propose a new input device, similar to a mouse that would have "breaks" on its roller. This would be to indicate the stickiness of the applied medium through force-feedback). In so doing, I presume, he anticipates a deeper engagement between the artist and the medium and from this, he

hopes, will stem a more resonant work.

Cockshot goes on to defend simulated paint, and its apparent correspondences with real world mediums by stating "to the artist user it is the apparent, not the actual correctness of the visual characteristics of the medium that is important". If *apparentness* is what matters, he is claiming his images *may as well be* real paintings. If this is so, or if this is how they are intended to be regarded, then they will be judged as such.

To summarise, it is very unwise for a non-photorealistic renderer to attempt the simulation of real painting and drawing for two reasons:-

Firstly if NPR is thought to be "attempting painting and drawing" it will obviously draw the same critical fire suffered by real painting and drawing, which may not be wholly appropriate. The view I am advocating here is that NPR image ought not be "misread" as *intended* paintings and drawings; they ought be regarded as something different.

Secondly, that trying to convince the viewer that what is seen is, or has been, a real, human-executed image, is an unnecessary burden for NPR to shoulder. In actual fact, as I will show, many of the qualities seen in painting and drawing can, in fact, be used with complete impunity to generate images of potential merit and the charade of simulating all aspects of painting is totally unnecessary and a barrier to the development of a new medium.

In Chapter 6 "NPR's relationship to other media", I discuss the inner nature of this new type of imagery and clearly define *how* it is separate to real painting and drawing. Before that, I would like to clarify how NPR images can force this separation from real paintings and drawing in order to become something in themselves.

## 5.2 Avoiding accusations of simulation through use of transcendent qualities found in hand media

The initial problem is to convince a critical community that NPR is not *necessarily* painting simulation – that it can be something else even though the evidence to date seems to point to the fact that NPR singularly aims at producing painting and drawing-like images. It is my aim here to show that in *mature* NPR this similarity is only incidental and not of central aim.

It is my conjecture that current *immature* NPR (most of the work to date), while possessing the

elements needed in the mature form, over-burdens itself with certain problematic aspects of which it must be purged. Certain aspects seen in current NPR can *only* be regarded as painting simulation – these are the *immature* elements. The mature aspects, while sharing certain similar qualities with painting and drawing, are *not* painting simulations, but are qualities which transcend any particular medium. I would now like to identify these problematic elements in order to help eliminate them from future development.

Certainly, conventional art and design media have qualities that are absent from computer graphic media. Many designing NPR solutions appear convinced that computer graphic imagery ought to try, somehow, to attain these qualities. Unfortunately hand media have so many different aspects, that identifying suitable candidates for abstraction and appropriation by an NPR system is a large part of the problem. My argument is that Natural Media Simulation arises (immature NPR) when the wrong ones are seized upon – those where the characteristic phenomena can not be transferred to the computer.

To a large extent this problem is phenomenological. If we are to attempt a transference of things from hand media to the digital arena we ought to ensure that the things transferred are not stripped of any essential phenomena in the process. An example of successful transference from the corporeal world to the virtual world can be found in word processing: the words from the physical page are transferred to the screen and memory of the computer. While it can be argued that something vital has been lost through the automation of scribing [Gaur 1987], whether a sentence is hand-written on paper or is composed of glowing phosphor dots on the screen, the most important phenomenon associated with writing and language (i.e. its linguistic content) has been maintained intact.

However, some things do not fair so well in this transference. For instance, food does not keep its essential phenomena (i.e. edibility) when transferred to the computer, and it is not simply a matter of increasing the accuracy of the representation. It can be clearly seen from the examples of non-photorealistic systems discussed how this philosophy has been disregarded, particularly with reference to Strassman, Cockshot and many of the image production strategies associated with the Fractal Painter program. Within these systems certain things presented as output are not the actual but the simulated – the action of chemicals, environmental conditions, the bumpiness of the surface and the trace of the hand – these things are not present, they are simulations of something actually absent. The essential difference between what can and cannot be transferred lies in the computer's ability to represent certain things. Let us consider the computer simulation of a storm; while this may prove useful for several reasons, the simulation is not intended to convince the user that the

storm is actually real and happening somewhere inside the computer. The storm is symbolically represented, not physically recreated, so most of the aesthetic experience associated with bad weather is lost. Words, as in a typed document, are already symbolised, so do not lose out so much when communicated via a computer.

We can apply this analysis to conventional media to see what aesthetic aspects and essential phenomena, remain intact and those which lose out when recreated on the computer. Take, for example, the wobbliness of a line – a wobbly line generated by a computer is no less wobbly than that generated on paper. The computer line wobble is not simulated (which suggests a removed experience) so much as actually present. However, take the bumpy surface of an oil painting or the trace of the artist's hand – these phenomena cannot be recreated algorithmically, they can only be simulated; a computer line may be *really* wobbly, but a computer generated surface (using today's technology) cannot be *really* bumpy.

We can present a catalogue, though probably an incomplete one, of various aspects of natural media that may be initially thought of as candidates for appropriation by the computer medium. This, in turn, can be divided into two; those which lose their essential phenomena and those which remain intact when recreated in the digital domain.

#### **Aspects of reflective substrate media that lose their essential phenomena when transferred to the computer**

- Chemicals and chemical behaviour
- Specific artefacts of physical tools (because they weren't actually used)
- Surface tactility - bumps and stickiness, runniness, non of these things can be genuinely represented by a computer.
- Trace of the body of human labour (when no body has been involved in totally automatic rendering techniques)
- Environmental conditions

The use of these aspects can ONLY be interpreted as natural media simulation because they only truly belong to natural media.

Aspects of reflective-substrate media that can be presented via the computer without recourse to simulation

- Colour, tone
- Shape, line, form
- Ambiguity
- Tentativeness, incompleteness
- Complexity
- Arbitrariness
- Evocation, poetics, associations
- Evidence of process as a time-dependent phenomena

These aspects can be used by an NPR system with complete impunity – i.e. their use does not constitute natural media simulation because these aspects do not belong to any specific media.

We can see from the two lists how some past attempts have run aground by attempting to appropriate aspects from the first list for integration into a computer system, and how well chosen goals might result in a promising system while avoiding a lot of unnecessary programming. Implicit within the second list is perhaps the entire thesis – that these are the qualities a non-photorealistic representation really aims at and can use with impunity because they transcend the specifics of any particular medium. These qualities are expressions of some transcendent referent, to which natural media executed images, and to which computer generated images can equally refer. These qualities cannot be privy to natural media alone; there are no reasons why they can not be successfully executed on a computer. These transcendental qualities are those outlined in the second list.

This stratagem potentially liberates the marks and gestures made in NPR from the grip of natural media.

### 5.3 Summary of Prescriptions for Mature NPR

Firstly the name "non-photorealistic" is very unhelpful as photorealism and non-photorealism are not supportable, distinguishable concepts. A new name will probably be forthcoming – I would



prefer Interpretive Rendering as it indicates the rupture interposed between the model and the image. However, for pragmatic reasons I have used the term non-photorealistic rendering throughout the rest of this thesis.

Real improvements in the quality of NPR images will be made by adding new techniques based on fuller description of the thing rendered. A raw digitised image of the sort used by many current NPR systems can not alone provide enough information. Included in the representation of the world (the source data) must be descriptions of geometry, material properties and other perhaps more subjective associations. Through the utilisation of this data the renderer will be able to increase the significance of the rendering style. Systems ought also to embody much more culturally specific, explicit descriptions of rendering techniques of the sort seen in many drawings and paintings (cross hatch, heavily stylised or characterised techniques).

While many problems are redressed by the further development of these techniques it is almost inevitable that the images generated through these techniques cannot approach the visual prowess, articulation and meaning associated with real paintings and drawings. This seeming impasse has been a demoralising factor in NPR research. However the real problem lies not in NPR's inability to match the quality of natural media but the assumption that NPR is *intended* to match real painting and drawing. This is a mis-assumption, NPR ought to be regarded as something in itself which has incidental similarities to painting and drawing but which does not seek to be regarded as painting or drawing.

This incidental similarity to paintings and drawing in natural media arises from deep cultural precedents which see painting in natural opposition to photography. However it has led several proposed and actual systems to date to take the comparison between NPR and painting (or more accurately the *metaphor* of painting and drawing) too far. This has resulted in the inclusion of unnecessary, damaging aspects in such systems. I have called this flawed approach "immature NPR". These immature aspects are those things which are distinct simulations of aspects that can *only* belong to painting and drawing – namely, the simulation of the trace of the body of labour, the simulation of chemical behaviour and physicality, and the suggestion of the physicality of the "canvas". Images which contain these sorts of deceitful simulations will always be compared directly with traditional paintings and drawing rather than as a new medium in its own right.

The solution to this is to remove these problematic aspects from future development leaving a purified mature medium. What is left is a medium possessing qualities, which we have initially seen in painting and drawing, but which transcend any specific medium. Ambiguity, tentativeness,

complexity, process, arbitrariness can be exploited in NPR. These qualities have been long desired in computer graphic rendering techniques.

This will result in images that are not visually the same as paintings and drawings, nor fulfil the same purpose nor occupy the same cultural domain. I am confident that given time, NPR images must be regarded differently to real painting and drawing and discussed in terms of their own merits. As such non-photorealistic renderings may be part of a new representational paradigm.

In the next section I will clarify further formal distinctions between "mature" NPR and real painting and drawing by pointing out certain underlying differences in the relationship between the respective image and the world it seeks to represent.

## Chapter 6

### NPR's relationship to other Media

I have so far indicated two ways in which current NPR could be greatly improved, by generating more "significant" techniques and, perhaps more significantly by avoiding the trajectory of natural media simulation. As a consequence of this one might ask if NPR is not a simulation of real world painting and drawing, and something unto itself; what actually is this new type of imagery and what is it trying to achieve? Obviously this type of imagery relates to real painting and drawing in some way, or these previous confusions and misnomers would not have arisen.

Here a formal comparison between various representational systems is presented as a series of schematic diagrams. From these several fundamental differences between NPR and real paint and drawing can be seen. Furthermore these types of schematic representation help distinguish photography from painting ; a task which traditionally has been fraught with problems. The schematic also shows how interactive techniques differ from non-interactive techniques.

We start with an analysis of the relationship between the thing rendered (the signified) and the rendering (the signifier). Within renderings of any type, be it by computer or hand, there is always a tension between "what is really out there" and what is rendered in the image. In real-world painting and drawing there is an overriding tradition of respect for the unknowable subjectivity and complexity of the world we inhabit; that it appears to us through filters and veils, and that we can only interpret and abstract. Added to this incoming data is all the pre-knowledge and prejudices of the artist, which cannot be disentangled in the final analysis from the incoming data; as a consequence, what is seen is "half there, half made-up" [Loyal 83]. The production of a drawing or painting , because it is done slowly over time, introduces the notion of feedback as a further complicating factor. In the production of a painting, for instance, we can represent this association between the world/artist/image as in Figure 14a.

This view of the world might be called the "fuzzy data" view – that incoming information is always shifting, infinitely complex and ultimately flexible in its interpretation. However within digital rendering or image processing the "world" has already been completely mediated through the choice of source data input to the system – either it is a model or a digital image, it has already been abstracted, it is much less open-ended. For the moment at least, underpinning all non-photorealistic conclusions are inevitable objective calculations based on this "hard data". With non-photorealistic rendering then, the rendering process approximates an inverse situation to that

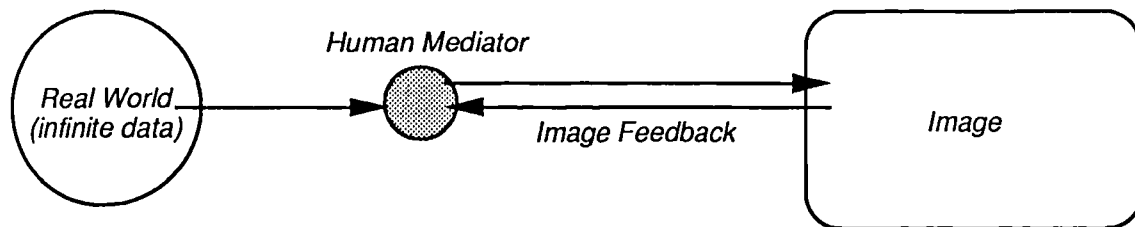


Fig. 14a  
Data flow diagram of World/Artist/Image relationships

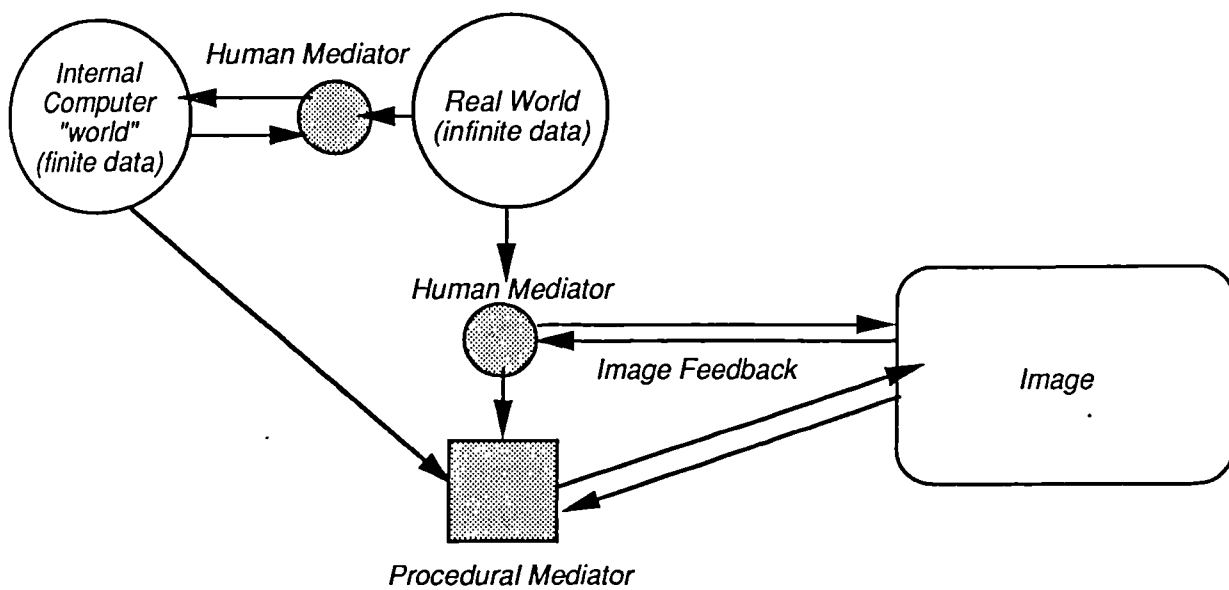


Fig. 14b  
Data flow diagram for Interactive NPR

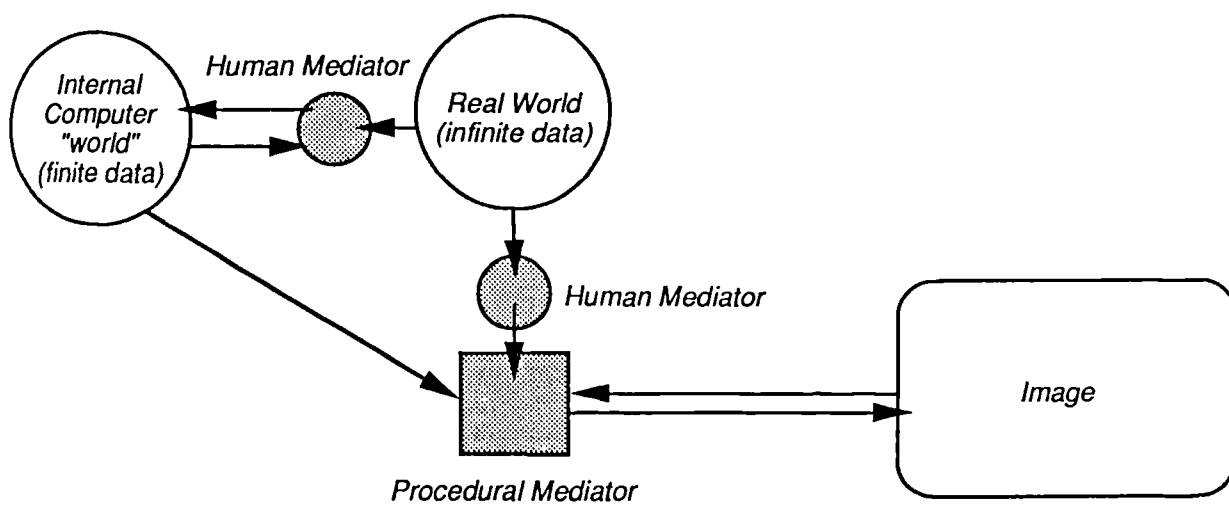


Fig. 14c  
Data flow diagram for non-interactive NPR

of deriving representations from the real world; in order to bring to life the limited, hard data available to the rendering system, we interpose extra levels of complexity. These extra levels of complexity imposed by an NPR system replace, to some degree, the complexity absent in the actual computer model. As such an NPR system becomes a "procedural mediator" between the computer-world (the hard-data model) and the "perception" of the model (the output seen by the user).

In the case of real-world rendering, rather than imposing new levels of complexity, we must resolve the incoming fire-hose of "fuzzy data" by filtering a limited number of aspects from the infinite. In this way the process of rendering in an NPR system can be regarded as the inverse process to that of real painting. This is illustrated in Figure 14b, which schematises the relationship between the world/artist/image for a time-base interactive NPR system.

This is somewhat simplified if we consider a non-interactive NPR system [Figure 14c]. This has many more discrepancies with "real painting"; it can be seen that the diagram for real painting bares little resemblance to that of non-interactive NPR.

These fundamental differences between the process of real painting, Photorealistic rendering and NPR tell us again why NPR must be thought of as separate to these things. Only when these differences are acknowledged will a fuller understanding of what NPR's cultural role and potential be formed. Only then will NPR carve a cultural space of its own as distinct from previous forms of representation.

## Chapter 7

# Developing an NPR System

### 7.1 Summary of Specifications and Prescriptions Made to date

The recommendations made for so far have been based on a critique of current systems arriving at an insight into what NPR could be in its mature form. To recap, it makes three broad prescriptions:-

- The improvement of data available to the renderer
- The improvement of techniques for increasing signification within the image
- The abandonment of natural media simulation techniques

Further insights into NPR emerged from the practical investigation which was conducted between January 1990 and January 1993. Several complete systems were assembled on various platforms; these are documented in the final part of the thesis. Over the length of the investigation certain important shifts in attitude and aims took place. The programme began by regarding NPR as an alternative to PR in terms of its ability to automatically render scenes. As such, one of the goals was that it would form the basis for animation techniques. Later the investigation switched from being conducted in the context of a Digital Arts faculty, where much of the output was video based, to that of its application to Architecture specifically, where the output was more hard-copy based. This had the effect of concentrating research on still image production for quality Architectural Presentation.

Following this shift, a few important changes to the eventual goals and techniques used in the system emerged. These mainly relate to interactivity, image resolution and compatibility. While resolution and compatibility are areas dealt with by all CG system developers and would be superfluous to this specific investigation, the issue of interactivity raises some very fundamental issues unique to the field of NPR.

## 7.2 Interactivity vs. Batch Rendering

So far this thesis has considered, and prescribed methods for generating NPR imagery without drawing a clean distinction between interactive and non-interactive techniques, the principal defining area of discussion has been the type of imagery produced (and its relationship to paintings and drawing), rather than the specific method of production. Interactivity, in the case of all image making is defined as the opportunity to modify the image while it is being constructed, as opposed to the approach which uses the longer feed-back cycle of setting parameters, rendering test images and observing results as its chief method of making the final image. It can be seen that the systems and publications to date propose solutions some of which are non-interactive (Yessio, Sasada, Mezei, Barnsley, Saito, Free Hand Plotting Group, the KATI system) and some of which are interactive (Strassman, Cockshott, Haeberli, Fractal Design Painter). Generally, the non-interactive systems are regarded as "renderers proper" and the interactive systems more akin to traditional painting systems.

However, it is the belief of the author that the most valuable results, in terms of NPR rendering, will emerge when a system is developed that uses both non-interactive rendering techniques along with interactive techniques. This is largely based on an investigation presented in Chapter 8 of this thesis.

In the final analysis interactivity is a very important issue – its inclusion in a system will have far more impact on the types of image produced than any amount of algorithmic automation. An image in which human judgement and intervention played an important role will always be better, in qualitative, aesthetic terms than those which generated through a purely procedural, batch method of production.

There are many reasons why this is so. Firstly the feedback cycle for parameter adjustment is much more rapid in an interactive situation – the same image, throughout its manufacture can be iteratively erased and re-rendered, rather than entire images being generated and considered in isolation, leading to a discontinuous modification process. Interactively produced images are much more complex and multi-layered for this reason alone; they store a record of their own history of modification and improvement, rather than presenting something which is thoroughly "correct", from which all the "mistakes" have been erased. Most importantly, the complexity and articulation available from human interaction far outstrips any amount of "intelligent", computer based process currently available.

There is another strong argument suggesting that NPR renderers would benefit greatly from the inclusion of interactive techniques. The community of designers, with which this form of rendering seeks to be accepted and affiliated, actually *enjoy* making images, and this enjoyment stems from deep, active participation in the process of image construction and the process of discovery brought about through experimentation. There seem no reasons for denying this pleasure.

The batch, procedural method to NPR has arisen from considering NPR as being an alternative to PR; in essence the automatic generation of images from predefined parameters and data. This approach is wholly desirable for animation purposes, because it enables the complete re-rendering of scenes with small parameter modifications between frames (for instance, moving the eye location). However, larger parameter changes between scenes usually require more human intervention (tweaking) in order to secure a good result. As has been stated, with PR rendering, a designer may spend a long time adjusting the eye point, perspective transforms, lighting and material properties before a satisfactory result is obtained. The criteria for judging the final result is usually simply by eye – there is little evidence to suggest a computer alone is able to select good eye points and environmental conditions for generating consistently pleasing results. As a result, while PR is often thought of as being wholly automatic, the user actually spends enormous amounts of time adjusting the end result.

While it would be desirable to speed up this feed-back cycle, PR renderers may not wish to be particularly interactive during the actual rendering of an image because human intervention would disrupt the expected unity and consistency of the photorealistic effect.

The desire for animated effects and photorealistic consistency has steered PR rendering techniques clear of render-time image interaction. However, the situation is different for NPR. Certain investigations into the animation of sequential NPR images has revealed some fundamental problems – the use of randomness in the construction of the image leads to overall inconsistencies between frames. This type of temporal aliasing is known as "boiling" – large parts of the image seem to seethe and the result is so visually distracting as to be untenable. Also, image which look convincing when taken individually often lose their desired quality when animated. Experiments in animating NPR imagery, its problems and solutions are areas for further work. The general feeling of the author is that unless the issues surrounding still image generation are properly resolved then animation problems cannot be resolved either; this thesis is mainly dedicated to an investigation of the generation of still imagery.

Added to the problems of animating NPR imagery is the fact that the unity and consistency often



desired in PR images are not a concern for NPR images. One of the engaging and attractive benefits of NPR is the ability to generate images which use many different representational schema within the same image. A very obvious example of this is Vignetting (where the edges of the image are rendered using less detail than at the centre) or the use of different techniques to indicate different materials as in drafting. Consequently NPR images are very often fractured and inconsistent in their appearance.

Consistency between frames in animation and photorealistic unity, then, are the two main reasons for the lack of interactivity in PR renderers. Neither of these criteria relate to an NPR system. Unless rapid speed of production of images, deskilling (cheap labour) and animation are part of the underlying rationale of a NPR system, then there are no reasons why NPR systems should not be fully interactive.

It is not the desire of the author to produce a system for deskilling the community of users by proposing methods for rapid and automatic image production. While systems for producing these sorts of images will undoubtedly emerge over the next few years (this is the stated aim of the KATY renderer [Glazzard 93]), as a painter and designer I am much more interested in the computer as a medium for communication rather than as a labour or cost-saving device.

To date, systems have been available which either expect the user to generate the work from scratch (painting, drawing, modelling systems) using complete interaction, or ones that generate the images through predominantly automatic techniques (PR renderers, post-processors) using no run-time interaction. What is being proposed is a system sitting between these two poles using "Interactive Rendering ". One of the most interesting and fruitful areas for future research will be investigating this new and complex situation. Images will be produced partly by the system, and partly by interaction; investigating this new form of collaboration between man and machine and attempting to form the correct synergy is beyond the scope of this thesis but is a very necessary direction for further work.

### 7.3 Consultations with Architects and Designers

This shift in direction towards an interactive solution, and many other modifications towards the general aims of a system emerged in part from discussions with Architects and Designers. This consultation was fundamentally important if the results of the investigation were to be anything other than an academic exercise. Consequently, the attitude and desires expressed by Architects,

Designers and Theorists towards NPR were gathered over a period of roughly two years as an integral part of the investigation.

Among the people consulted early on were:-

- Masinori Nagashima, Infomatic Inc (Tokyo)
- John Hare and Steve Miller, formerly of Arup Associates, Now Miller•Hare, London
- Ernie Lowenger of River CADS (Splinter of Richard Rogers), London
- Simon Ruffle, Andrew Coburn of Cambridge Architectural Research, Cambridge
- Roger Matthew, Scott Brownrigg and Turner,
- Nicholas Ray, Nicholas Ray Associates, Cambridge

Later as the program developed demonstrations of prototype systems were held along with slide shows of more advanced imagery. Among those consulted at these later stages where

- Caroline Brown of Sir Norman Fosters and Partners, London
- Steve Jones, Architects Journal, London
- Intergraph User Group, London
- William Mitchell, Dean of Faculty of Architecture, MIT, Boston MA
- GDS user group, London
- Dr. Patricia Wright, MRC Applied Psychology Unit, Cambridge
- Ed Hoskins, Hoskins Associated, London
- Dr. Kathy Carter, Xerox EuroPARC, Cambridge

The initial approach to the consultation was to show architects and designers some early attempts at NPR. These were generated, initially by techniques described in chapter 8, using the "Expressive Marks" procedural renderer. Later, interactively produced images using the "Magic Painter" and a very early version of the "Piranesi" system were used. One limitation was that there was no opportunity for Designers to use the system themselves throughout the length of the investigation due to the fact that the prototype systems were too difficult to use, non-robust and implemented on a specific platform (Silicon Graphics) using in-house software. However, the final system "Piranesi" presented with this thesis is, at the time of writing, close to being a generally usable piece of software.

As a consequence of this limitation, feedback on the response to the images was gathered from slide

shows or from controlled demonstrations of certain techniques. Listening to the objections and enthusiasm expressed by the consultants, it was possible to compile a list of potential benefits, and applications for a final system. The results of the consultations are split into three sections:

- Expressed enthusiasm over why non-photorealistic rendering would be a good idea, what it would mean to their practice.
- A wish-list for a "dream-system" with no technical considerations outside the purely aesthetic/ergonomic domain.
- A harder definition of a complete system based on several demonstrations of prototype systems. i.e. an Application overview

## 7.4 Benefits of non-photorealistic rendering

### Interpretation and Feedback

As indicated in Section 2.3 of this thesis, photorealism presents particular problems when used throughout the design process; this is due to the implicit insistence of *completion* even when the image may be showing a partially completed scene. As such, photorealistic images resist conceptual mobility and are difficult to "think around". A simple photoreal CG scene is not evocative in the same way as is a simple pencil sketch; it provides no conceptual or emotional space into which we can fit our own complex experiences and so flesh out the lack of detail with our memories and associations.

For this reason PR is unpopular during the early, formative stages of the design process, and shown with great caution to the client, usually only when the design is well underway and most issues have been resolved. NPR would allow CG and CAD techniques to participate more in early stages of the design process by providing methods for making images from simple models, created in the early stages of the design, that would be suitably vague or tentative. Each rendering could be slightly different from the last, further suggesting the degree of surety of the design. This would enable CG imagery to participate in the design feed-back cycle in a more positive way, and may in time become an inspirational tool equivalent to the "pencil and paper napkin".

### Tentative Modelling

The ability to represent different parts of the model in differing stages of completion and certainty

is made possible by integrating into the modelling system a means for tagging parts with a "tentativeness" description. This could be done by the user or perhaps by the machine, which would monitor the amount of modelling done in relation to some pre-set notion of completion - (e.g. have the nature of the walls and ceilings been fully described?). The representational nature of tentativeness could be decided and tailored by the user. One designer might simply choose fuzzy or wobbly lines, another might choose an effect that renders differently each time.

### Incomplete Model representation

Not only does this approach lead to acceptable images early on in the design process, it also suggests that the final model might not need to contain the same level of detail as a photo realistically rendered model in order to achieve similar levels of visual acceptability. Imbuing a model with the final level of detail required for respectable photorealistic images is usually one of the most time consuming aspects of modelling. Railings with finials, decorations, trimmings are fussy to model and add frustrating amounts of memory to the final file, making manipulation increasingly laborious. An entourage of trees, people, clouds present particular problems to photorealistic computer graphics rendering as yet not fully overcome. Heavy stylisation or photomontage techniques are often used to avoid these limitations.

Non-photorealistic rendering offers the alternative of merely suggesting detail. For instance, the capitals of Corinthian columns might be regarded as a particularly difficult modelling problem – swept solids and parametric patches could be used with considerable expense to time, processing power and memory – and yet they can be pleasingly rendered with a few deft strokes of a pen. Clouds, grass, trees and people can be similarly simply represented; the visual effect may well be more pleasing than the exhaustively detailed photorealistic equivalent.

Through omission of surface detail an image can gain clarity; we often find ourselves more impressed by an economy of representation than by any amount of *trompe-l'oeil* finery. With current visualising systems and rendering approaches there is a seemingly limitless requirement for detail - the visual appearance of the objects purport to be as ideal as their mathematical representations and we feel persuaded by such thinking to describe every hair [Kajiya 1989, Watanabe 1989], every blade of grass [Reeves 1983] as an individual and highly scrutable entity. A blade of grass, a strand of hair need not be a many-faceted object or dynamic particle system - it can be a simple mark or subtle gesture.

Consider an image of a field of corn. A typical photo-realistic approach would be to have each strand of corn as an elongated, articulated prism for a minimum representation of a cylindrical

surface off which light could be reflected. Using a non-photorealistic approach each piece of corn need only be mathematically represented by a small set of connected line segments. This greatly reduces the modelling required. Each notional strand of corn is fleshed out along its length with a previously defined mark. The human eye is very forgiving when viewing unashamedly roughly worked, gestural renderings. Questions of accuracy of detail become irrelevant. This leniency is not enjoyed by photo-realistic interpretations. By using an expressive renderer not only is the necessary modelling much reduced, but also the general level of demand for surface detail is lessened. The image is generally less fussy, allowing the user to work more quickly, or spend our time resolving other aesthetic issues.

While NPR can present the complex with simplicity, it can also represent the simple with complexity through its ability to offer high levels of detail almost for free. Sky, water, terrain such as grass, mountains, rocks, and many building materials can all be represented using relatively simple configurations of marks, with no or little input from the model.

#### **Integrating hand and machine Images**

Modifying Non-photorealistic rendered images by hand, either within a paint box type program, or on the actual hard copy is less problematic than attempting the same with a photorealistic image – some of the most satisfactory images could be a fusion of hand drawn marks overlaying a non-photorealistic image.

#### **Solving Traditional Montage Problems**

In terms of generating a convincing PR montage, a good deal of work usually has to be done to the image in order to ensure that the computer generated model “sits” in the image convincingly. Reflectance mapping, palette adjusting and distance fading are the most commonly used techniques in this process; these can be computationally expensive and are by no means a guarantee of success. If however the separate elements of the montage are rendered using NPR techniques then combining these elements ceases to be an issue. Disparate elements are unified through the application of the same style or they may be rendered in completely different styles with impunity as overall pictorial consistency is no longer critical to the image’s success.

#### **Varying degrees of realism , altering modes of representation**

Perhaps one of the disadvantages with PR images is the tendency to present scenes which are completely uniform in their representational devices; that the edges of the image are rendered with the same degree of visual accuracy as the intended focus of attention. With hand-renderings

there is always the inevitable inconsistency between separate parts of the images, resulting either from a desire to emphasise some parts over others, to suggest differences in materials, or as an accidental effect of human inconsistency. Either way the image gains a richness which is lost to photorealistic imagery. An NPR system may well be able to produce images ranging between the photorealistic imagery normally associated with CG to any degree of sketchiness or arbitrariness. Moreover it would be able to use these varying degrees of realism within the same scene. This capability could be used to emphasise certain parts of the scene, which could for instance be rendered in full photorealism, while the surrounding context could be rendered in decreasing degrees of realism, fading out to the slightest suggestion at the edges of the image.

### **Summary**

Interviews and discussions with architects and designers have generally indicated that the integration of Non-photorealistic rendering within art and design practice may well help reposition the role of CAD and CG in several positive ways. It is felt to have implications on the Designer-Client relationship, the design process, modelling issues, and perhaps most importantly a possible cultural repositioning in terms of the aesthetic range, capability and prowess of computer generated imagery.

## 7.5 A wish list for an NPR system

Having been presented with a selection of possible rendering approaches and the benefits mooted the consulted Designers were then asked to flesh out the more specific capabilities of the system. This was without consideration to more practical issues, such as cost or computability.

- Rendering styles are to be as diverse as possible ranging from the simplest sketch through to very heavily worked images bordering or achieving photorealism where required
- All the techniques of photorealism - shadows, reflections, surface textures, participating mediums, bump-mapping, complex shading (from radiosity for instance ) must have their equivalents within the system
- The rendering style is to be independent from the modelling - e.g. complex models can be rendered as sketches and simple models can be rendered in very complex ways
- Styles must be very user editable, enabling the user to invest worthwhile effort in the pursuit of individualism and excellence
- Techniques are to be collected in extensive libraries
- The system must be easy to use, using standard interactive techniques for the definition of rendering parameters. i.e. rapid testing and modification of styles, no written scripts or typed numbers
- The system must integrate with current CAD and Computer Graphic systems, it must take as input standard model formats (e.g. DXF) and output standard image file formats (e.g. TIFF, PostScript)
- Advances in Architectural modelling must not be excluded, rather enhanced.
- The system must respond to all features present in a model; geometry, lighting, material attribute

- Images must be produced relatively quickly, e.g. one finished image per hour
- The ability to overwork/rework images over a period of time
- Images must be very high resolution, if required. The system ought to aim at images around the  $4000^2$  pixels and so working via rapid scrolling or zooming would be required
- Rendering methods are to be both interactive and automatic
- The interactive part of the program must include all the standard features of a painting package (e.g. Photoshop), these include cut and paste, selection and masking, filtering and interactive painting tools and Undo operations

This wish-list was used to envisage the skeleton of a potential real system. The following chapters describe several experiments underlying the systems Expressive Marks and Magic Painter which, in themselves move towards a understanding of how many specific parts of the final system, Piranesi, are implemented.

## 7.6 Application Overview

The Application overview presented in Figure 15 describes the component parts of a fully functioning Interactive Non Photorealistic Renderer (INPR). This is an admittedly vague description of the final Piranesi system and has been compiled in hind-sight, but it can be seen in the following chapters how aspects of the various attempts at implementing NPR solutions fit within this framework.

The modelling and viewing application has undergone several changes. Initially it was a non-interactive modelling and viewing library written in C by Jason White and Julian Saunderson at Middlesex University. This has subsequently become the VYNIL system. Later it was the interactive MODEL system developed by Tim Wiegand and John Rees at the Martin Centre in C++ on an SGI platform. The 2-D scene representation used by the NPR system has changed significantly over that period.

The definition of rendering styles by the user and their application to the scene is perhaps the



largest problem for consideration. Initial experiments using the Expressive marks system proved inflexible in the long run. Object orientation used later allowed for a much more flexible authorship of techniques.

The description of the technique to be applied to the scene and its final application by the system are of course interlocking problems. In order to understand the final visual appearance of a technique the user, currently, has to understand the method by which the computer executes certain operations. Hence the use of OO techniques in the final Piranesi system has not only implications on the definition of style, by making them more flexible, but also on the execution of styles.

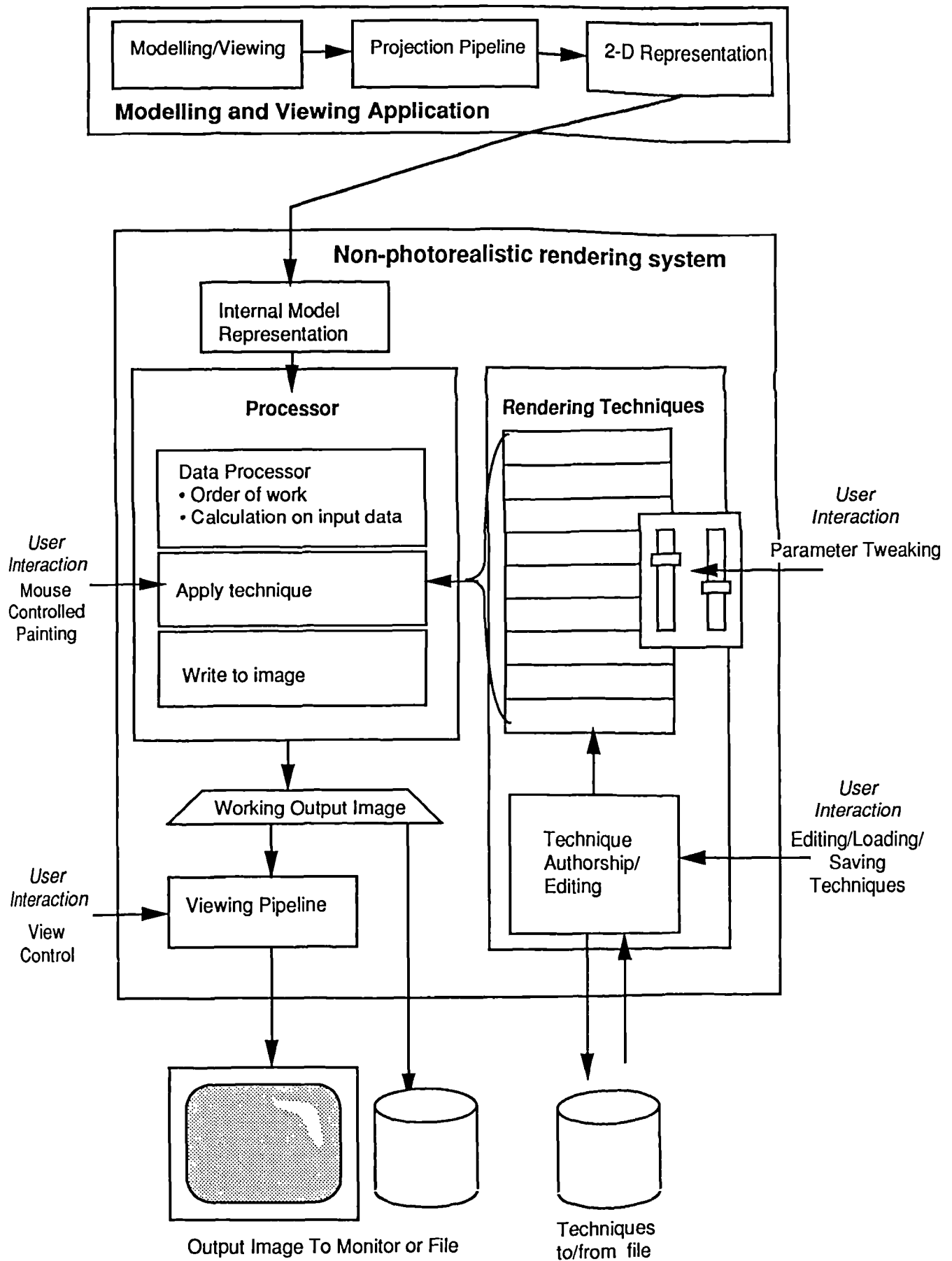


Fig 15. Application Framework for an Interactive NPR

## Chapter 8

# The Expressive Marks System

### 8.1 Introduction to the Implementation

A prototype system was first implemented between April and September 1989 and documented in the MA thesis [Schofield 89]. Despite vast changes in the techniques and overall aims of the system, certain features developed in this first implementation have endured. This system is primarily an engine for generating painterly marks along the lines of Strassman [85] and Haeberli [90]. These are then used to "render" 2-D and 3-D geometric models in a way which is still novel. The initial, and subsequently abandoned, premise held throughout much of this implementation was that images were to be produced with no run-time interaction – i.e. that the system presented in this section is wholly self-sufficient in the generation and placement of marks within a scene. This necessitates several innovations beyond the work of Strassman and Haeberli who both use semi-interactive systems with mice or pressure sensitive pens as input devices during image generation.

The nature of the NPR, in this implementation, is presented as a two stage problem – mark generation and mark placement in response to user-defined pre-sets and some internal representation or model to be rendered. A large part of the "Expressive Marks" system concentrates on the efficient modelling of visually rich marks. This follows my initial belief that a lot of image quality comes from the quality of the marks composing the image.

The Expressive Marks system was originally written in C on a mainframe VAX with output to an IKON Frame store. Later this was transferred to SGI Personal Iris with GL and later re-written in C++.

### 8.2 The Mark Synthesiser

#### The Concept of the Mark

The General Mark Synthesiser is envisaged as an engine for generating a wide range of efficient, parametrised marks that can then be used by other modules for specific purpose. Independent of placement within the scene the parameters of a mark are broken down thus:

- **Colour** - the overall colour of the mark. The overall chroma (RGB components) of the mark is assumed to be fixed throughout its length
- **Transparency** - this is important for the effective layering of marks to build local texture. Transparency may be set overall. It is also automatically varied over the length of the mark to create fading and building, and over its width to give various hard-edge/soft-edge effects. Transparency can be varied on a pixel level stochastically to produce a chalk effect. A more sophisticated chalk-effect would use texture Look-up Tables (LUTs) to generate larger-than-pixel sized irregularities. This has not yet been implemented.
- **Size** - width of the mark
- **Shape** - a simple envelope defines the overall shape of the mark suggesting change in pressure over execution
- **Path characteristics** - length of mark, general direction, curves and wobbles

This approach caters for two distinct types of marks: **linear marks** (like pencil, pen and brush marks) and **stipples** (like daubs and blotches). Three other fundamental types of graphic elements are necessary to provide general functionality; these are detailed later. (Section 8.5 Summary of Required Graphic Elements).

An alternative approach to synthesising marks is to sample them from the real world; this has been shown to be successful by Haeberli [90]. Extending the sampling concept to include some degree of real-time control over the marks is a worthy area of investigation. Problems such as the interactive resizing of sampled marks, stochastic perturbation and the type of looping used in sound sampling to generate samples of different lengths from the same source data could be explored.

#### How Marks are made - pixel by pixel

The general model for a linear mark is based on the commonly used technique of overlapping discs of colour applied along a path at some regular spacing [Rivlin 86]<sup>1</sup>. These are written into a frame buffer and the display updated whenever convenient<sup>2</sup>. The disc model has the advantage of accommodating a path that changes direction throughout its execution. There are no difficulties

---

<sup>1</sup>A detailed chronology of paint systems is given by Robert Rivlin in "The Algorithmic Image", Microsoft Press 1986. He details systems from Sutherland's seminal Sketchpad, through Shoup's generalisation of the raster-based painting system [XeroxPARC 72-73]. Most extant systems use the circular uniform stamp as the primary drawing tool

<sup>2</sup> The display is updated at the end of every complete mark, so they "pop" onto the screen *in toto*. This has proven perfectly adequate, even for interactive painting

negotiating corners and no possibility of stroke italicisation because the disc can be moved in any direction on the image surface and its profile remains the same. However there are some inherent disadvantages due to the discs overlapping. The amount of overlap set is quite critical; too much and unnecessary overwriting of pixels results, too little and mark starts to break up or look "beady". Overwriting is ordinarily avoided by implementing the Posch and Fellner algorithm [Posch and Fellner 89]. However this is only suitable if a brush is completely opaque - i.e. a solid colour.

### **Modulation along Mark Length**

The modulation of intensity or pressure over the run of a linear mark is controlled by an "envelope" similar to that used in music synthesis [Chamberlain 80]. This object returns a pressure value for any point along a mark (defined parametrically) and, at its simplest, is initialised by defining a mark's "attack, sustain and decay". A mark composed of 50 overlapping stamps would have perhaps 0 pressure on the first increasing to a pressure of 1.0 at its 10th stamp, maintaining a pressure of 1.0 until the 30th stamp and then tailing off to 0 at the 50th stamp. Such a mark would be defined:

```
mark.setEnvelope(attack = 10, sustain = 20, decay = 30, maxPressure =  
1.0);
```

This simple envelope specifically aims at modelling the nuances at the beginning and end of strokes. In natural mediums it can be seen that pencil marks usually have a brisk attack and long decay, drafting pens have a very steady pressure throughout avoiding line intensity variation, while etching and gouging tools usually display a slow attack and harsh decay. The attack and decay phases are independent of mark length, while the sustain phase enters into a loop for longer marks and defines the maximum intensity written [Figures 16 f,g ]. A more sophisticated model for the envelope is adopted later in which any number of phases can be set up. This considers the repeated coupling of two parameters – rate and level – to define a particular phase in the envelope and is similar in this respect to digital music synthesis techniques. This model allows for much more complex mark characteristics. The ability to loop and jump between phases based on simple rules may be a worthwhile extension.

### **Modulation over the Width of a Mark**

In order to create interesting internal variations across the width of the mark, colour intensity can be varied (i.e. opacity) as a function of distance from the stamp centre. By changing the opacity in respect to this distance, and by thresholding at varying values soft and hard edged effects can be achieved. More importantly, this enables the writing of stamps of subtly varying sizes. The

function of intensity/distance from centre to pixel can be modified to create linear, inverse square, inverse cubed relationships resulting in gradual or sharp fall-offs in value. This process is made more efficient by pre-calculating distance of pixel from centre as a look-up-table. The final amount of pigment written to the image is calculated thus:-

```
finalStrengthOfPixel()
{
    //i.e. centre pixel -> 1, edge pixel -> 0
    dist = 1.0 - distanceFromBrushCentre / brush_radius;

    return colorAmount( envelopePressure, dist);
}

colourAmount( pressure, dist)
{
    amount = pressure*pixelContact;

    edgeEffect  = amount*softEdgeAmount + addFlattenOut( amount );

    return  final_amount = multiplyByChalkEffect( edgeEffect );
}

addFlattenOutEffect( product )
{
    if( product > cutOffThreshold) return 1.0- softEdgeAmount;
    else return 0;
}

multiplyByChalkEffect( amount )
{
    return amount * (1.0 - chalkEffectAmount * random(0,1));
}
```

The changing size of the stamp, and the fact that stamps are spaced at even intervals along the length of the mark, results in small stamps at the ends of the marks becoming separated from the mark as a whole. This can be avoided by bunching stamps as a function of envelope pressure. There are other unwanted artefacts caused by the overlapping disc approach; the overlapping of translucent stamps causes lens- shaped artefacts to appear in the mark. Air-brush algorithms avoid such artefacts by only applying a tiny amount of colour per stamp, hence the overlapping build up of colour is fogged out. This amount of over-writing is unacceptable for efficiency reasons.

A solution to both these problems is to replace the disc-shaped stamp with a pre-oriented thin "wiper". The wiper is essentially a rectangular shape and is oriented orthogonally to the direction of travel over the surface. This acts as an alpha mask through which the disc-shaped stamp is

pushed. The wiper is defined in a cardinal position as 3 columns of pixels, the centre column being set to write the corresponding disc-stamp pixels unchanged, while the outer ones are used to soft-edge the wiper effect by having the alpha set at 50%. After the direction of the mark has been specified a tilted version of the wiper is made by calculating the rotated elements in another buffer. This is done with floating-point accuracy to partially avoid rounding errors. The wiper is then marched along the mark length using a digital differential analyser [Bresenham 65] to calculate its position on the output image and the stamp is written through the wiper. The three column approach is found to be good compromise between speed of writing and getting rid of residual rounding errors. Without this soft edging bands are seen to appear across the mark at certain critical angles. This method is generally faster than the pure disc-stamp approach because fewer pixels are overwritten in the length of a mark, particularly in wide brushes. 'Streaking' is also achievable over the length of the mark by setting internal values of the wiper to different alphas.

Writing thick lines via the application of rectangular or skewed brush heads is not new, and much work has been done to ensure clean behaviour when such butt-ended lines join with each other at angles. One approach is to put a circular fillet at the intersection [Foley&VanDam 90: 962-964] or mitre the join. However this only works easily for lines where the overall colour is uniform. When more complex internal characteristics have to be maintained such joining techniques do not work so well. At present the wiper approach does not accommodate re-orientation of the writing head throughout the execution of one mark, although this has not proven to be a significant problem.

### Mark Path

Linear marks contain several parameters appertaining to the default path they will make over an image, where no extra information is forthcoming from some other controlling mechanism. All marks have a default direction or "bias" in which to travel, defined by a two dimensional vector and a default length. Added to these are two other parameters relating to how straight the path is: amplitude and frequency. These are handed to a Brownian noise function<sup>3</sup> which returns a wobbly path of the correct direction, length, frequency and amplitude. The general method for travelling along the path, in keeping with the DDA, is a uniform step between two points, x1y1 and x2y2 thus:

```
mark_length= sqrt ((y2 - y1) * (y2-y1) + (x2-x1) * (x2-x1))
dx = (x2-x1) / mark_length
```

---

<sup>3</sup> Brownian noise is the product of a cumulative function where results have a drunken walk of set amplitude and frequency. Such functions therefore need to keep track of their last result in order to calculate the next.

```
dy = (y2-y1)/ mark_length
```

the Brownian noise function return a cumulative displacement orthogonal to the direction of travel

```
displacementX += randomGauss(amplitude, frequency) * (-dy);  
displacementY += randomGauss(amplitude, frequency) * (dx);
```

where the Gaussian random number generator returns numbers  $\pm$  some amount around zero [Figures 16a, 16b].

### Stipples

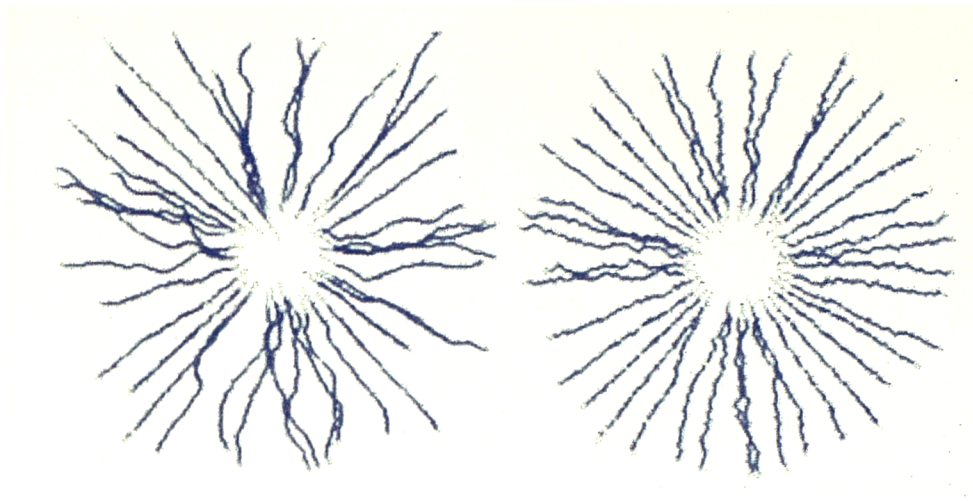
One of the reasons circular stamps remain attractive is that they can be used very effectively to generate stipple-marks by distributing stamps around a notional centre rather than along a path [Figure 16c]. A Gaussian distribution is favoured for controllable scattering. The probability of a stamp landing near the notional mark-centre is set to be larger than it landing further away; by adjusting the standard deviation of the gaussian curve the distribution of stamps around the stipple centre can be altered. Rather than associating the envelope value with the mark length it was associated with an individual stamp's distance from the stipple centre and imposed rather like a jelly-mould over the top of the whole stipple. Usually when the distance is 0 the envelope is set to full pressure. Stipples do not suffer from the internal artefacts of linear marks because their comparative lack of order hides any systematic artefact.

### Mark Variables and Mark Instances

A method for varying every new instance of a mark around a user-defined ideal is necessary for generating the types of subtle differences seen between one mark and the next in natural media. This is formalised by coupling every parameter governing the mark with a sister parameter governing the variation around the ideal. There are two methods - one for floating point parameters, such as cut-off and chalkiness, and one for integer parameters such as RGB values.

```
//for floating point calculations  
instanceVal = idealVal ± random(-1,1) * variationVal;  
  
//and for integer calculations  
instanceVal = idealVal ± random( -variationVal, varaitionVal);
```





*Fig 16a and 16b  
Varying the amplitude and Frequency of Linear Mark Paths*



*Fig 16c Stipples*

*Fig. 16d Mark blending*



*Fig. 16e  
A mark with high variability applied many times*



*Fig. 16f*  
*Various modifications to the Envelope of a Mark*

The programming concept is within the object-oriented paradigm, wherein a mark is an object possessing initialised parameters (member variable state) and specific functionality (methods). The mark object is actually a mark *generating* object, which can be requested to write marks to the image. This object may be very long-lived in terms of execution time and responsible for the painting of many visual marks on the canvas. Within a mark object parameters are stored in the *MarkStyle* and *PathStyle* structures (more objects). These each hold the *ideal* and *variance* parameters necessary to generate a third *instance* set of parameters which define how the next visual mark will actually look [Figure 16e]. Long-lived changes in the parameters for a mark are thereby affected by altering the ideal and variance data, and short-lived mark modifications – ones that only affect the next mark – are brought about through modifying the instance data after it has been initialised. Therefore these structures are made accessible (public) for parameter modification. A schematic representation of the mark object is given in Figure 17.

While it is necessary to define all the ideal mark parameters it is not necessary to define all the variations - if one is undefined then the ideal parameter remains constant. The final parameter set for the Mark Synthesiser is actually quite large and the current mark complexity could be reduced with little effect on image quality. Both in execution and interface the parameters governing the mark effect, (i.e. the type of instrument used) are kept separate from those governing the path of the mark (i.e. where the instrument goes). The X Windows resource database is used to hold the data [Nye 90]. This ensures that sensible defaults can be set up so that only parameters of interest need be re-worked for new marks. Modifications to marks can be read in from a series of files each containing only a partial amount of information.

### Mark Blending

A useful method of mark parameter control is through blending between two pre-defined marks parametrically. All linear-marks and all stipple-marks have the same parameter sets. It is therefore easy to implement blending between two similar objects [Figure 16d]. This is particularly useful for creating gradated effects often seen in drawing, such as lines that fade with distance or a range of marks defined between wobbly and straight. The function is interfaced thus

```
blendedMark = mark1.blendMe(mark2, amount);
```

### Run-time Interface to Marks

While the mark object is fully independent in terms of its own graphical capabilities, typically some other object controls and places the mark within the scene. This "controlling object" may vary any of the mark's parameters in response to information about the scene. For instance, when



# A graphical description of the Mark Object and its functionality

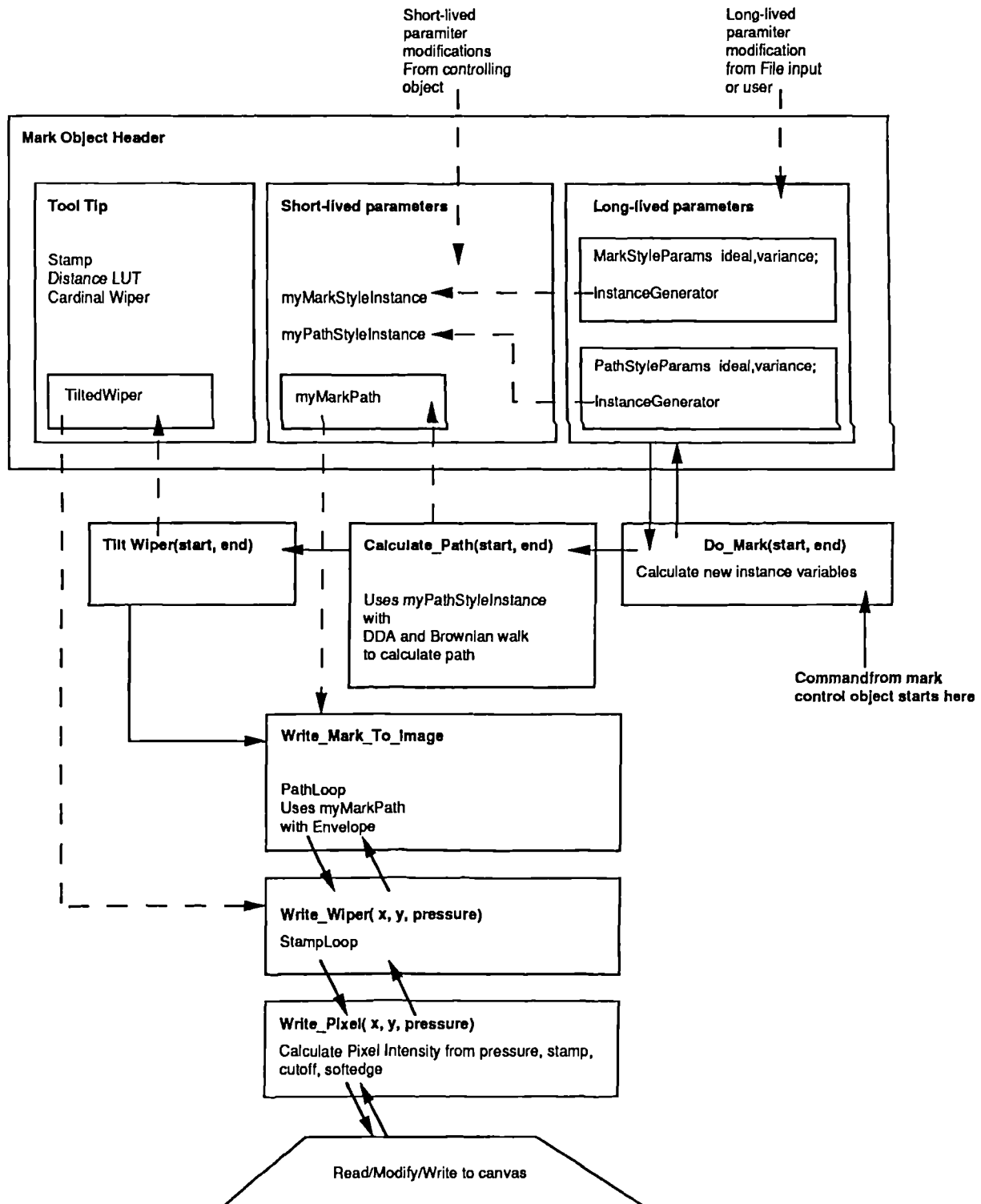
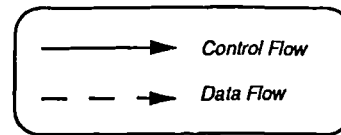


Fig. 17 The Mark Synthesizer object structure

“cloning” a frame buffered image the mark has to be recoloured according to the corresponding sample point in the buffered image – this would require the short-lived colour instance parameter to be modified before writing a mark. When drawing geometrical shapes derived from a model, the mark's short-lived path start and end points are replaced with those of a given line-segment.

Marks objects are defaulted for a minimal interface. A command to place a mark at a certain point would result in the default length and orientation parameters being used. Three main functions place marks on the canvas:-

```
//use default direction and length
mark.markPoint(Point2d startPoint);
// i.e. use default length
mark.markVector(Point2d startPoint,Vector2d direction);

mark.markAbsolute(Point2d startPoint,Point2d endPoint);
```

The final function was found to be the most useful and assumes that the intelligence of positioning marks lies outside the mark synthesiser module.

### 8.3 Marks used to Post-process Digitised Images

Two differing ways of utilising marks are initially implemented; the reconstruction of frame buffered images and the rendering of simple geometric models.

A very simple technique is used for sampling an input image for colour values, recolouring a mark accordingly, and painting a mark at a corresponding location in the output image. Even when using a consistent mark over the entire image surprisingly rich images are generated [Figure 18] . These marks may be applied on a scan line basis from top to bottom with enough random displacement for each mark to hide any underlying gridding in their distribution. However, a more visually engaging process is achieved by implementing a “random walk” application of marks<sup>4</sup>. This could

---

<sup>4</sup>The system was initially conceived with animation in mind and for creating dissolves between images where one image would “paint” itself over another. Even with a relatively slow system (i.e. 1 mark every ten seconds) this technique is quite mesmeric to watch. The image builds up slowly and at a certain point the viewer suddenly sees what is being



*Fig. 18 A photorealistic CG rendering post processed using marks*

be thought of as simulating some invisible artist's brush wandering over the surface of the image. In order to achieve good general distribution of marks over the image the system monitors coverage of the canvas to avoid excessive over-painting. If the random walk finds it can not progress due to areas already having been covered (for instance if it paints itself into a corner) it will jump to a more distant location on the canvas and resume.

Real improvements are made in image quality when automated mark orientation and mark clipping is implemented. Marks can be clipped to colour region boundaries depending on some threshold, further to that, marks "look around" a point to find good direction to go in. To do this the system surveys the source-image in a clock-wise direction around the sample point to establish whether a straight run can be achieved without transgressing significant colour boundaries in the source image. If the mark can not be placed *in toto* in any of the tested directions then the mark is shortened and another clockwise search executed. This enables marks to find good directions in even quite fiddly detail, as for instance in the astragals in the window of Figure 19a and the branches of a tree in Figure 19b. While reconstituting images via mark painting is now commonplace [Fractal Design 91] [Aldus Gallery Effects 92] the automatic painting mode, colour-clipping and best-direction finding are still unique to this system.

### **An interactive approach to post-processing using marks**

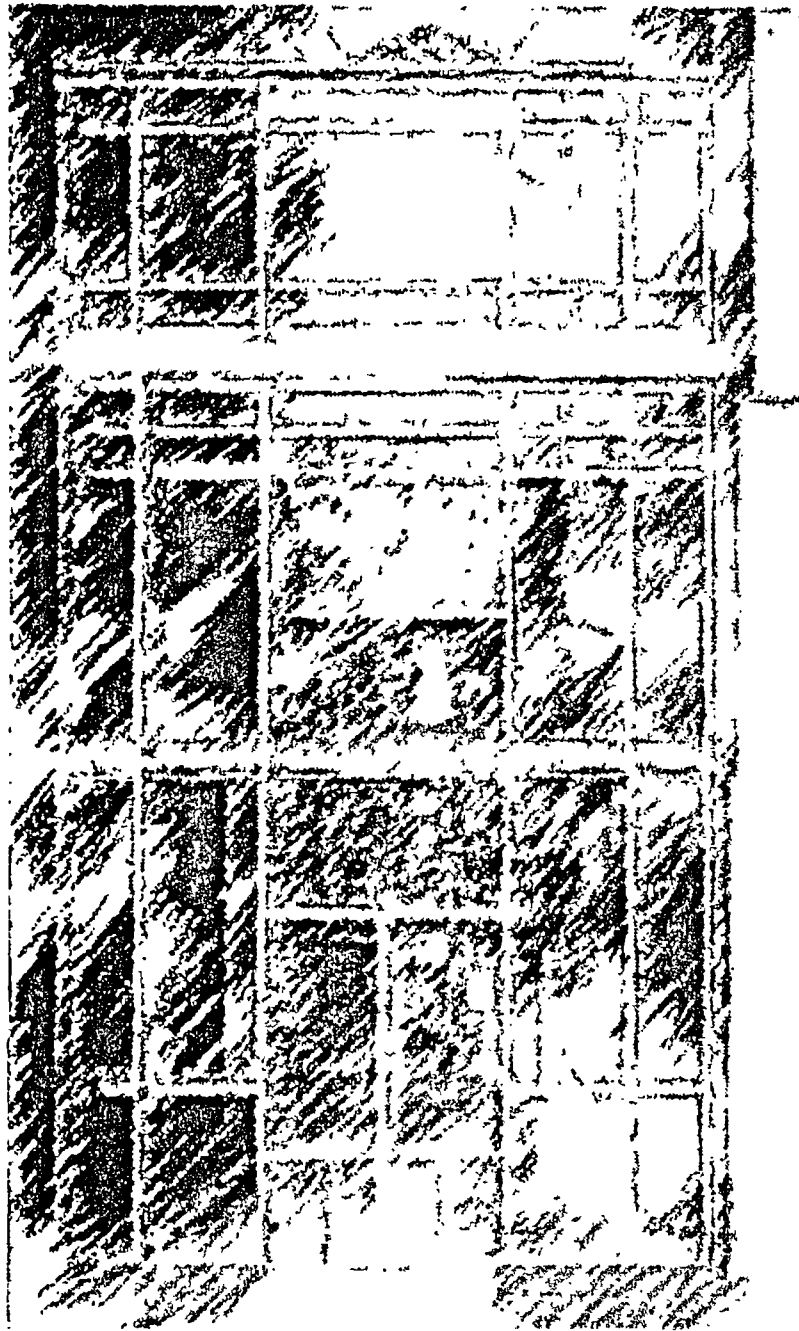
#### **Magic Painter**

This technique is extended through the development of a semi-interactive system. The problems of sharing the image-making work between the computer and the user are partially explored. To minimise the matrix for this experiment the type of marks available are limited; a pleasing chalky mark is pre-set, alterable in terms of its width, direction, length, opacity and wobbliness. Colour region clipping sensitivity is governed to by an "Ambiguity" slider [Figures 20a and 20b]

The user is able to guide the area being painted around the screen using the left mouse button. If the mouse stays still for any length of time the painting focus wanders off from the mouse pointer and, if left long enough, wanders randomly round the screen. This means that an image is painted even if the user does nothing. Stokes can be made to run in different directions and have different lengths by dragging with centre mouse button in the desired direction and length. An "erase" setting is implemented by colouring the current stroke with a uniform background colour. This gives the effect

---

painted.



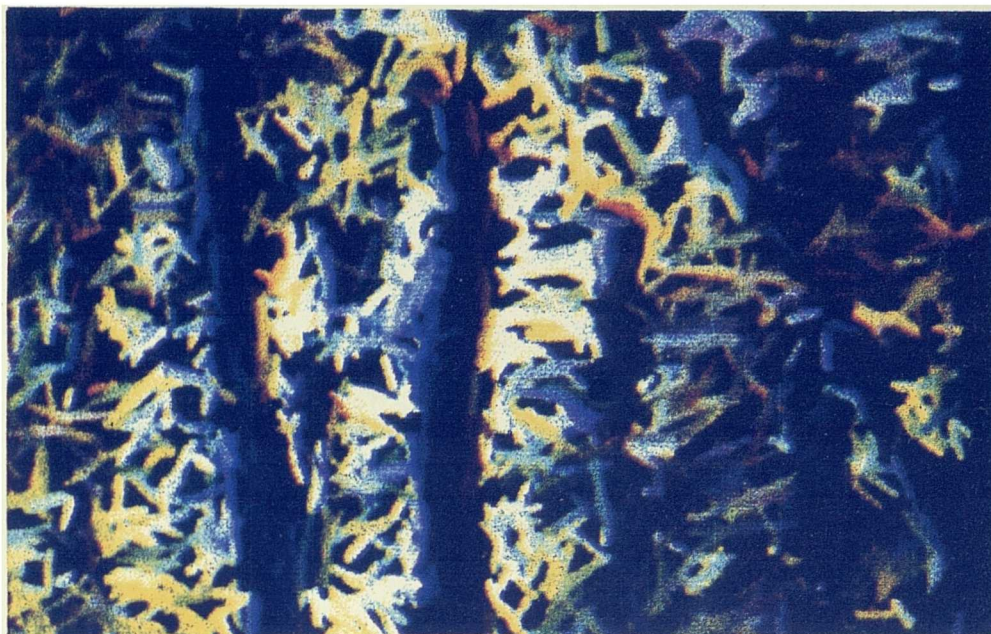
*Fig. 19a Post-processing with intelligent marks used to emphasise window outlines*



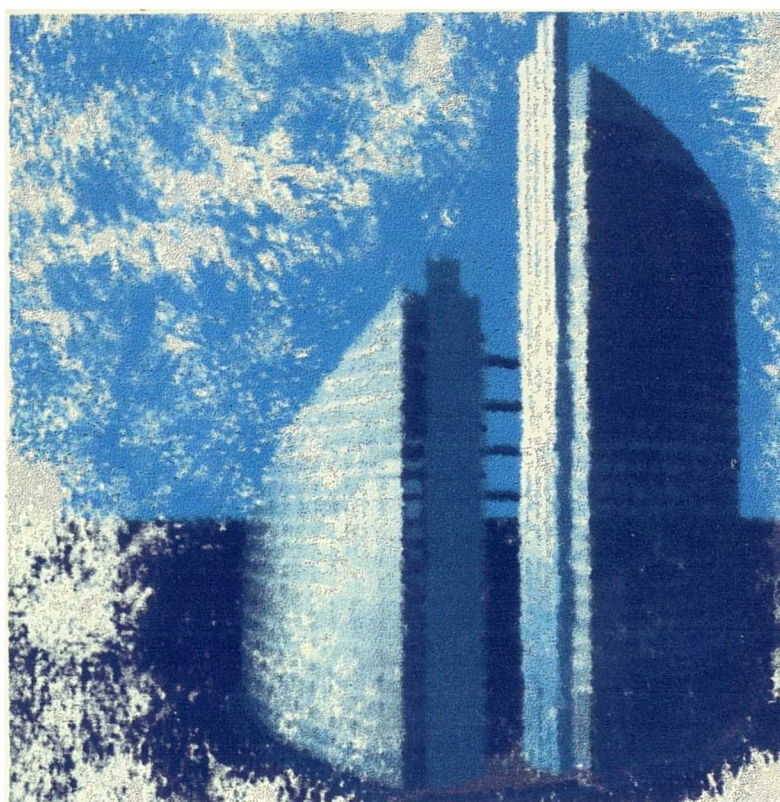
of rubbing out using a soft-edged chalky rubber and some of the most pleasing textures are found to be combinations of coloured marks and erasing.

While the system has a strong signature it is able to produce many pleasing types of image. Despite the successes and popularity of the system it is understood that on many levels this type of image construction is fundamentally banal. The system has been given the suitably twee name of "Magic Painter", and one can imagine it being popular with children. One of the most successful methods of generating an image is to leave the system running with a large mark setting for around ten minutes until it has wandered around the screen enough to give a very vague impression of the overall image, then interact with the system, bringing out areas of detail using a small short mark with low ambiguity setting. The smallest mark is able to resolve the image to near photo-realistic levels of detail; any level of articulation between the crudest blotch and the finest detail is achievable. *Vignette effects can be achieved by rubbing out the edges of a picture.*

In conclusions, the types of images produced through partial interaction are far superior to those generated by the machine alone. With some modification the "improvisational" aspect of the automatic painting could form a useful aspect of many paint systems.



*Fig . 19b "Intelligent" marks with image-sensitive orientation*



*Fig 20a An image made by Magic Painter*



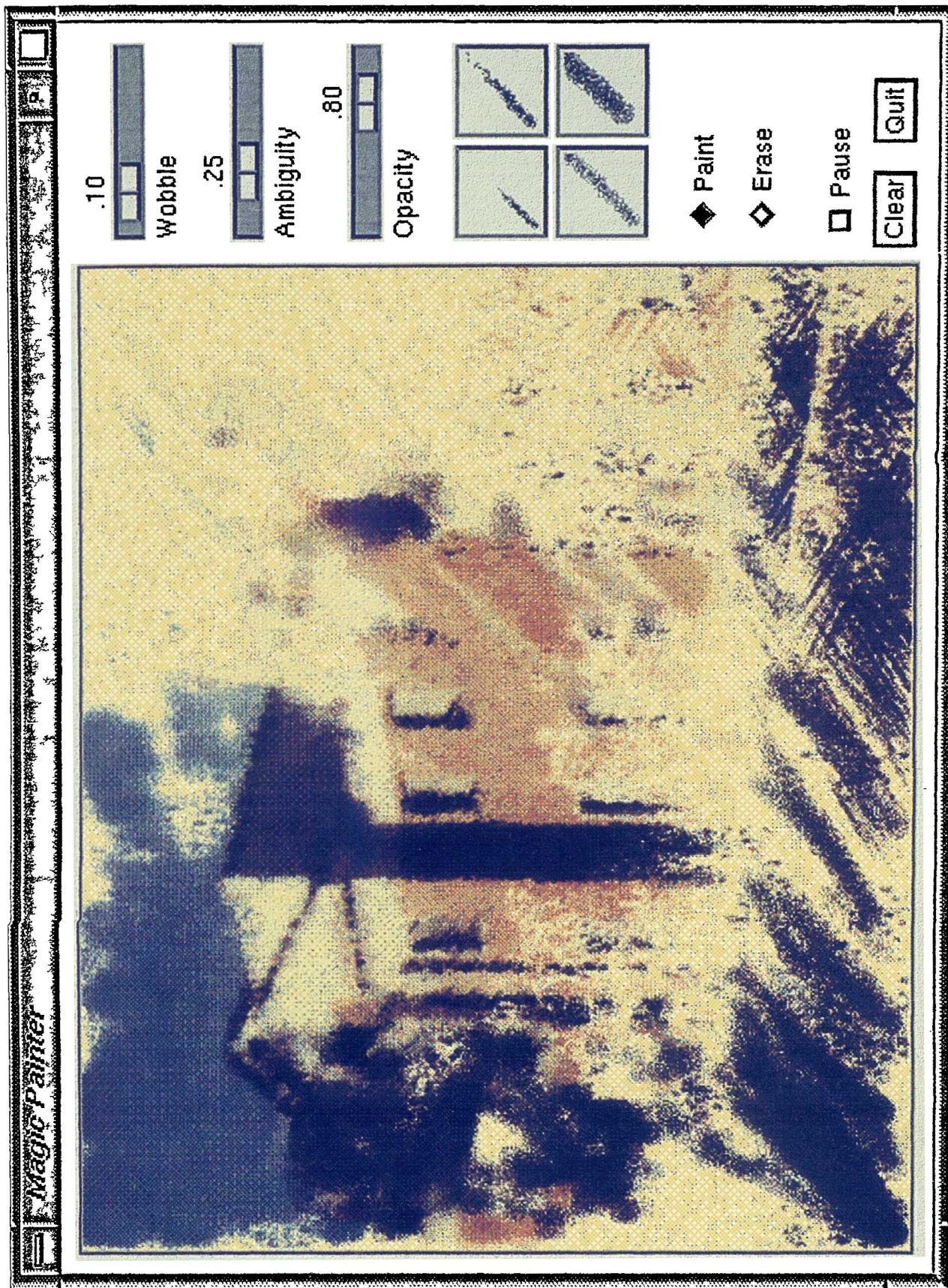


Fig. 20b The "Magic Painter" interface



## 8.4 Marks used to Render Geometries

### Drawing polylines

Within the paradigm of 2-D geometric rendering two distinct operations are catered for; polygon filling and polyline drawing. With each of these a large range of rendering styles can be defined. Let us first consider line drawing. To do this, a set of polyline vectors are scaled by defining the final drawings desired bounding box in terms of screen co-ordinates. The scaled vectors are then used to guide marks. The length and wobble of the mark is predetermined in the mark's PathStyle object but the controlling object specifies spacing between the start points of consecutive marks, which if set to low number results in overlapping. The random skewing of mark start and end points and an amount of cumulative displacement between individual stamps in the mark is used to generate a range of new line styles [Figure 21a].

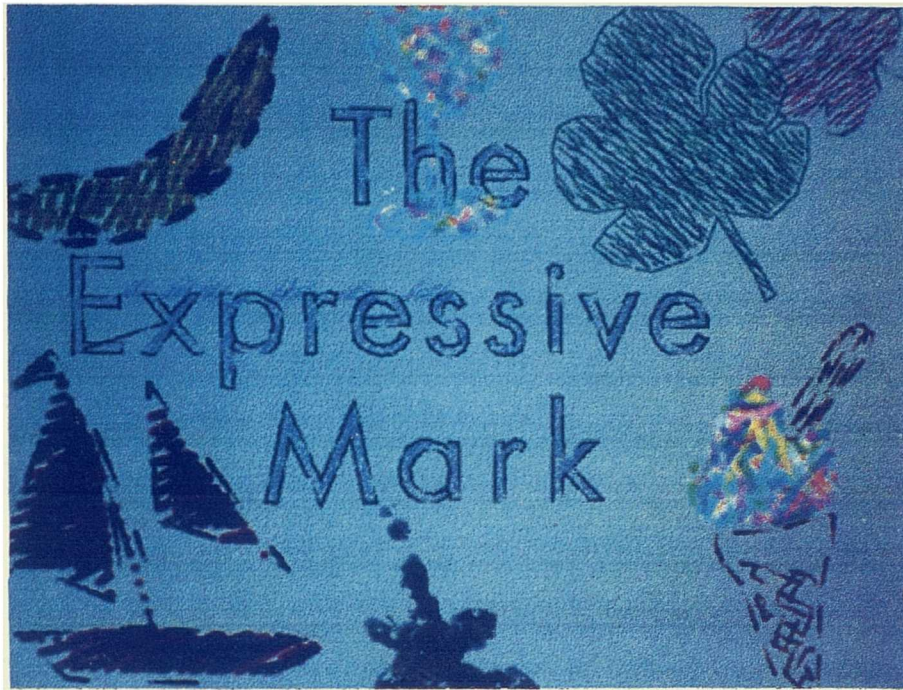
### Filling 2-D Polygons with Marks

The filling of 2-D closed polygons with multiple marks is initially achieved using a polygon filling scan-conversion algorithm [Foley&VanDam 90: 945-965] to step over the polygon in regular strides of many pixels in both X and Y. At each step in the scan-line a new mark is placed. This poses the problem of trying to abolish any visible gridding of mark placement which would look totally unnatural [Figure 21b]. The random displacement of marks using this method is frustrated by the fact that once a point has been established as being inside the polygon, to then randomly displace it by even a small amount in X and Y would possibly place it back outside the polygon.

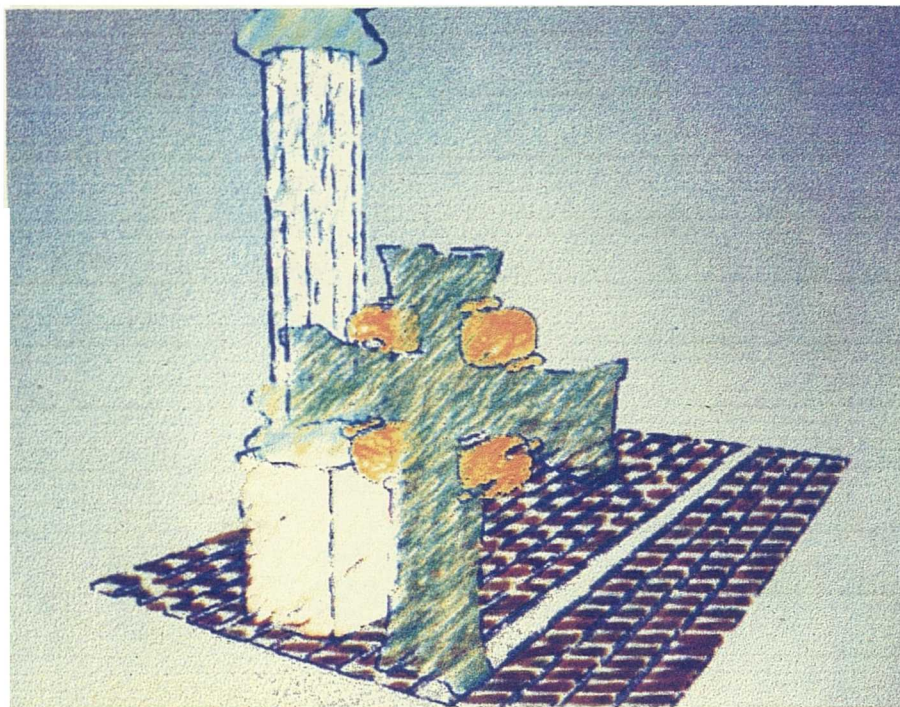
Displacements along the scanlines only (i.e. in X only) are testable for polygon enclosure and this is not sufficient to hide the regular spacing in Y. A better solution is to regularly step over a bounding box for the polygon in question, randomly displace the point in both X and Y, and then checked to see if the displaced point is still inside using the winding method [op cit : 965]. The amount of displacement ought to be roughly half that of the regular spacing distance, this being the minimum required to completely conceal the regular stepping process of the spacious scan line approach .

Not only must the start point of the mark be checked for being inside the polygon; every point along the mark as it is written (i.e. every new Wiper position) must be checked, to see when, or if the mark exits the polygon. This slows down execution, and perhaps a better approach would be to implement a binary, sub-dividing, search along the marks path to determine clipping points. This would be somewhat similar to the Sutherland-Cohen clipping algorithm [op cit: 113-117].

Apart from regulating the general density of marks by changing the scan line step, and changing the



*Fig. 21a*  
*different outline and filling styles*



*Fig. 21b*  
*Scan-conversion method used to fill projected 3-D shapes*

overall direction of all the marks within a fill, no other parameterisation has been implemented due to the prohibitive complexity of the geometric problems of dictating a marks behaviour within a closed polygon. Although even with this basic level of control the results are generally pleasing.

### **Rendering 3-D geometries by a polygon filling approach**

The ideal data required for rendering 3-D objects via this approach this approach is a set of perfectly tessellating polygons calculated from a 3-D model using a perspective projection algorithm, with all the hidden surface calculations resolved<sup>5</sup>. However, most viewing pipelines either project complete polygons (except for those clipped by the viewing frustum) and use a z-buffer [Catmull 74] to resolve overlapping and depth sorting or use a z-scanline approach when a z-buffer is not available [Bouknight 70]. In other words, they do not geometrically clip polygons against each other at any stage, and this poses serious implementation problems for pursuing this approach. The required type of projection –i.e. a Weiler Atherton projection [Weiler and Atherton 80] – is both difficult to implement and unsuitable for the reason that it triangulates the whole scene. This generates too many unwanted lines for edge drawing and poses problems for visually pleasing filling effects.

By way of concluding this avenue of investigation some half-measures are implemented which used back-facet culling and silhouette finding to reduce projected shapes to something approximating the desired input, although this approach can not be considered robust. Also, the general problem of depth-sorting is not addressed [Newell 72].

A partial solution to the problem of hidden surface removal can be attempted by painting the scene back to front based on 3-D bounding box mid-points. Before any polygon is filled its entire screen area is erased, either with a flood-fill or by using background coloured marks. The polygon is then painted using the final fill. This approach sometimes benefits visually from the residual traces of previously filled and erased parts of the image [Figure 21c]. However, successive rubbing out ,over-painting and often complete obliteration of distant objects results in a very slow execution of the complete scene. Also this type of crude depth sorting, which uses the centre point of a 3-D bounding box as its sole criteria fails to produce correct results more often than not.

---

<sup>5</sup> The VP (Vertex Polyline) system developed by Jason White and Julian Saunderson, Middlesex University, 1990, was used in these tests.



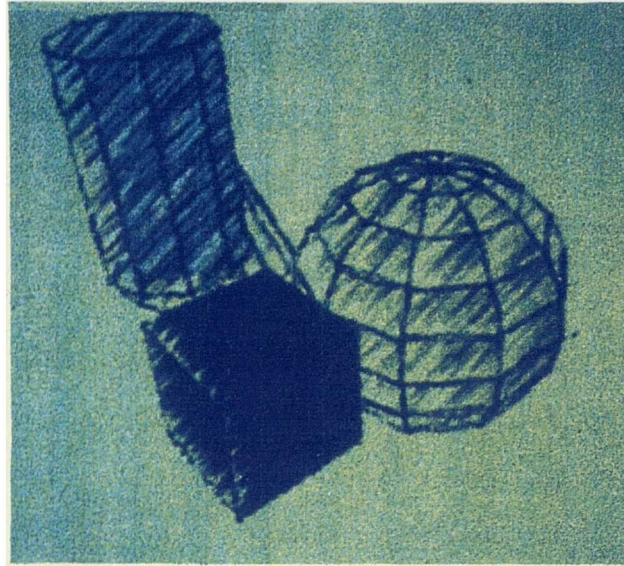


Fig. 21c  
*3-D projected geometry rendered using the scan-line approach*

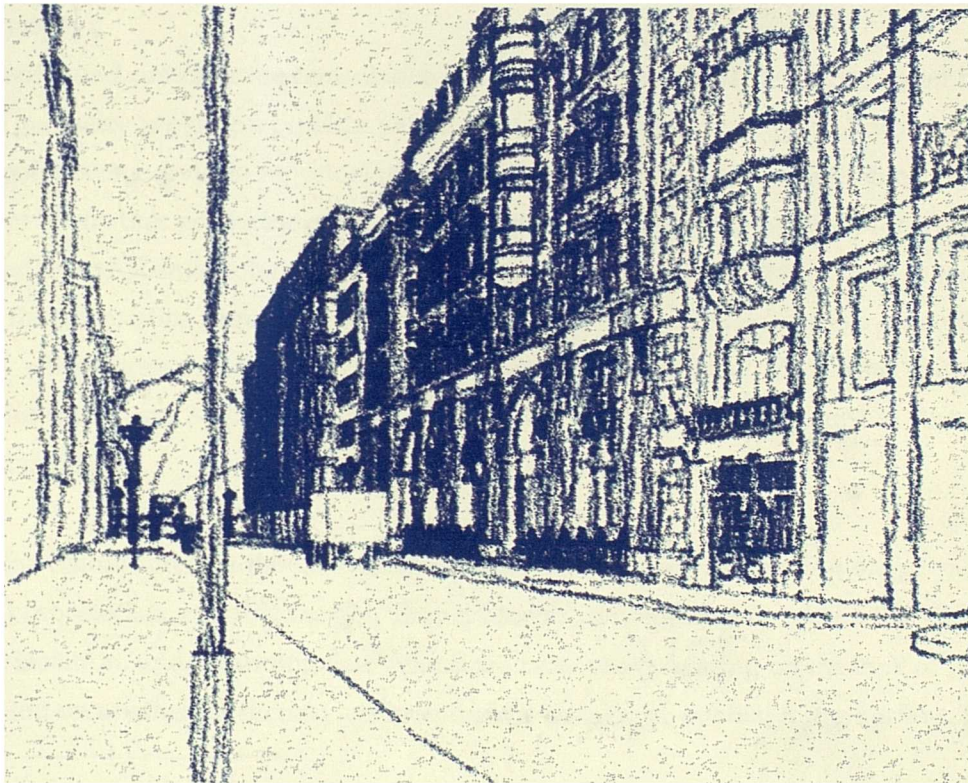


Fig. 22  
*A hidden line "plot file" rendered using marks (courtesy SBT)*

### Hidden Line Drawing

One of the most successful usages of this initial vector-based approach is in the rendering of hidden line drawings generated from 3-D models [Figure 22]. A hidden line drawing is stored as a "plot" file, and used instead of the standard polygon list from 3-D and 2-D data. Polygon filling is not possible using this approach due to the destruction of coherent polygons during the hidden-line calculating process although several solutions to this problem could be hypothesised<sup>6</sup>.

This method creates images not dissimilar to the Washington University "Wobbly Pen Plot Group", with the advantage that the type of mark characteristics are much more controllable. It also suffers from the same problems, namely the over-excessive bunching of lines in distant objects. Ideally, the Z dimension of the plot, which is often stored intact in some file formats like DXF, can be used in this situation to thin out lines according to some simple rules. An image-space solution, based on assessing and limiting the number of lines drawn in a certain area of screen space, may also be possible.

### Some conclusions

For a successful and complete implementation of the "Expressive Marks" polygon/vector based system the geometry projection pipeline would have to be re-written along the lines of a Wieler-Atherton polygon subdivision representation of the scene. It would also be desirable to leave Z values intact for further geometric calculations, inferring the specific geometric properties of parts of the scene through Z-interpolation schema over the enclosing polygon. Not only is this solution unsuitable for the reasons previously stated, it would also inevitably lead to slow execution, particularly where large models are used. Complex run-time geometric calculations to determine the original normals and 3-Space locations of points within the scene would further limit the speed of the application.

However, the realisation that the generation of non-photorealistic images is a staggered hierarchical process is a valuable insight. This approach, on one level considers the visual subunits in isolation, understanding that these units have heavy bearing on the look of the final rendering, and on another level considers how these sub units are applied to the surface. The final

---

<sup>6</sup> One approach is to infer what constitutes inside and outside polygons by giving lines more data relating to their initial relationship with the 3-D polygon. A new sort of normal pointing to the centre of the original polygon would facilitate this. Areas on the image would be searched for surrounding lines and some sensible fill calculated.



look of the image is a synergy of these two separate tasks and it is quite hard to disentangle and see the implications (on paper) of their separate roles. Indeed this is in fact a somewhat myopic view of much more elaborate hierarchy indicated in the analysis of drawing techniques in Chapter 4. The next section proposes a system for specifying a much more elaborate hierarchy of drawing definition and control functionality.

## 8.5 Summary of Desired Graphic Elements

When this investigation began it was aiming quite specifically at the modelling of painterly marks and, in the long run, paintings and drawings. Much of the implementation discussed thus far arose from this initial aim; i.e. the modelling of parameterised marks that possess complex internal structure, that could, if desired be set to be visually similar to certain natural media (charcoal, pencil, felt-pen, crayon, ink etc.). However, in the light of interim arguments developed thus far, this simulation of natural media has been abandoned. The new view favours much more arbitrary methods of image construction that are not beholden to the methods of natural-media image production. Consequently a whole new range of techniques for mark-making that are further away from natural media techniques have become valid techniques for appropriation. Apart from the Linear Mark and Stipple graphic elements, other fundamental graphic elements have been identified as desirable in a completed system.

### Area Texture

Initially it had been thought that this type of graphic effect would be manufactured by applying a field of smaller marks, but throughout the investigation it became apparent that it was difficult to achieve the desired effect and extremely slow. A gradated translucent texture fill of the sort seen in water colour, for instance, would be impossible to achieve using an accretion of small marks. Perlin [85] offers some simple methods for generating areas of textured colour. Noise functions of differing pseudo-random statistical properties [Knuth 69] applied to an image are valuable ways of introducing controlled or unexpected variations over the surface of the image. Other noise functions are spatially based and use combinations of sine waves. Textures can be given specific "edge-effects" to complement their graphic subtlety. Edge feathering and strengthening can be calculated reasonably quickly using slightly adapted flood-fill or "blotter" techniques. Directional feathering and strengthening would be desirable as this is often seen in many natural media techniques.

### **The dot**

Many NPR effects can occur at the scale of the pixel. On one level the pixel-sized dot can be regarded as a high-resolution painting device. Other more sophisticated techniques may be expected to work on a pixel by pixel method such as Re-rendering or noise functions, edge finding and hatching.

### **The effect**

This is a graphic manipulation element, probably larger than a pixel ( e.g. a circular brush type element), capable of performing various filters and effects to the image, such as Sharpen, Blur, Contrast enhance, Fade, emboss, add noise. More complex effects, which contain directional information are able to Smear and Smudge the surface. Many of these effects are available in standard painting packages [Photoshop 93]. An effect may also be applied through an alpha mask (a mask controlling the degree to which an effect is applied) to alter large areas of the image.

The easiest transformations to apply are one-to-one pixel transforms [Schumacher 90] resulting in the value of each pixel changing according to a simple rule that treats pixels in absolute isolation. Effects that change brightness and contrast, resulting in negative, solarised and posterised pixels can be generated. Careful solarisation results in some edge finding. More complex filters are required to consider individual pixels in relation to their surrounding neighbours. This approach in its simplest and most elegant form is known as a convolution filter [Foley and vanDam 90: 631-636] and surprisingly primitive relationships between adjoining pixels result in complex and non-intuitive effects. The most simple and most intuitive effect is perhaps an image blur in which each pixel under consideration is altered to be more like some of its neighbours. By averaging with all surrounding eight pixels quite a heavy blur is achieved, but by averaging only with those, say two or three to the left a type of motion blur effect is achieved. Embossing, sharpening and edge-detection can all be achieved using simple convolution filters.

## Chapter 9

# The Enriched Pixel Scene Description

The next three chapter describe the experiments which culminated in the final Piranesi<sup>7</sup> NPR system . This constitutes the first general framework for an NPR system, and as such is a significant contribution to knowledge. Most of the points and ideas detailed in this description have been implemented, while others have still to be finalised; these are indicated as areas for further research.

The description of the Piranesi System begins with the implementation of a radically different scene description technique.

### 9.1 The Enriched Pixel Image approach to rendering scenes

The difficulty and unsuitability of the polygon/vector approach of the Expressive Marks system suggests a radical re-thinking of the run-time 3-D scene-data interface.

The solution adopted by the Piranesi System is to use the conventional graphics pipeline provided by most graphics workstations to generate the complete scene as a raster image. Normally this *means only keeping the final coloured pixels of the lit scene as RGB values*. However other useful data can be associated with each pixel in a scene, such as Z-depth and material type. This is in keeping with the work of Saito and Takahashi [90] who store geometric information in a G-Buffer during rendering, but is extended to provide a more general solution. The resultant extended pixels or "Enriched Pixels" (abbreviated to EPixel), are stored as a raster image, these can then be saved to a file for later use. Figure 23 shows the structure of the standard pipeline and the points at which information is calculated for inclusion in the EPixel file.

EPixel images might reasonably contain per pixel :-

- Z depth of surface from viewer. These are derived from sampling the z-buffer or scanline-z

---

<sup>7</sup>The name is completely arbitrary but honours the first professional Architectural Draftsman Giovanni Battista Piranesi (1720-79). The word Piranesi also contains the letters standing for Interactive Non Photorealistic Renderer.

while rendering. X and Y world-space can be calculated from the inverse perspective projection if available.

- Surface normal. If absent this can be calculated from the surrounding world Z's.
- Material Attribute. This in turn may include references to the Layer of the material (e.g. ground, sky, external wall, interior floor) and the Material Type (e.g. rock, cloud, ashlar, marble). These material types could also contain references to Material optical properties. (see Header section)
- Shadow masks. These are derived from the lights in the scene. A mask states whether a pixel is lit or unlit by a particular light. The number of masks corresponds to the number of lights in the scene.
- Lit RGB values. These are calculated in the pipeline via traditional photo realistic techniques but could go if lighting, material (optical) properties and normals are available as a re-render can be achieved.

Other less conventional material properties may be included on a per-pixel basis

- Tentativeness of model. The "surety" with which the model has been completed may be included to influence the final style of rendering. This could be user defined or derived by automated monitoring processors in the modelling system.
- Subjective descriptions of elements within the scene. Old/new, like/dislike, cheap/expensive, melancholy/frivolous etc. These could be used to influence final rendering styles.

The header for such a file would include

- Information on the inverse perspective transform used. If the inverse perspective matrix is absent, re-world points can be estimated from the angle of view, if we assume The "Up" vector is up, line of sight is horizontal and the centre of projection is screen-centre.
- Light Table. For each light in the scene – position, polar distribution, RGB intensities, ambient conditions.
- Layer table. Reference from pixel to text string.
- Material Table. Reference from pixel to text string and to optical attributes LUT.

The first advantage of this method is that all the geometric calculations and hidden surfaces are calculated reliably and quickly by tried and tested techniques. It neatly separates out the modelling and scene definition process from the non-photorealistic rendering process. All modelling and viewing attributes can be dealt with by a completely separate system and the EPixel scene dumped as a file for use by the Piranesi system. The system used in the working implementation is developed around the SGI Inventor [Strauss & Carey 92] scene-definition library and the ACIS solid modeller [ACIS 92] with real-time walk-through ability and rapid shadowing. The "Model"

system has been developed at the Martin Centre, University of Cambridge by Dr. Tim Wiegand and John Rees. The interface was built using the XDesigner system [XDesigner 92] and the viewing pipeline was built around the SGI image vision library [Neider & Tillman 92].

The second advantage to this approach is that all the data for all points in the scene are constantly and quickly available. As a consequence the final rendering does not have to happen in any particular order (unlike a scan-line rendering approach). This implies that the system is well suited for run-time user interaction or multiple over-rendering.

The third advantage is that this approach breaks down the distinction between rendering geometries and post-processing raster images. This led to divergent solutions and functionality in the Expressive Marks implementation between rendering 3-D scenes and post-processing digital images. Piranesi system handles 3-D scenes and digitised images as though they are one and the same. As such it is a much more general solution to the whole field of non-photorealistic rendering.

This also make the implementation compatible with other experimental enriched-pixel research. Automated photogrammetry obtains depth information from digitised photographs by comparing stereo pairs, or through the use of range-finding apparatus [Beyer and Streilein 90]. Optical material attributes can be calculated from photographs by "de-lighting" them [Durisch & Anderheggen 91]. This data can then be used to determine the material type. Through the combining of Digital Photogrammetry with Un-lighting techniques it may be possible to generate EPixel images from photographs.

The disadvantage, apart from the general "cast in stone" feeling to this type of 3-D representation, is mainly that of file size and aliasing. Aliasing is caused by the quantisation of geometric and material information over the raster. While the z-values are stored with floating point accuracy resulting in accurate geometric calculation, the edges of regions between differing materials suffer from the "jaggies". Chiefly for this reason it may be envisioned that the final EPixel Image should also contain references to a hidden line representation of the scene – each EPixel would be able to locate surrounding edges as vectors in the hidden line description – enabling smooth drawings of the edges. The aliasing caused by the inaccurate quantisation of regions of one material against another could be redressed by the inclusion an A-buffer [Carpenter 84] which is used to define *how much* of a specific material is represented in each pixel. This may also assist the rendering of transparent objects within the scene, a technique for which the EPixel approach is generally unsuited.

### The Standard Graphics Pipeline

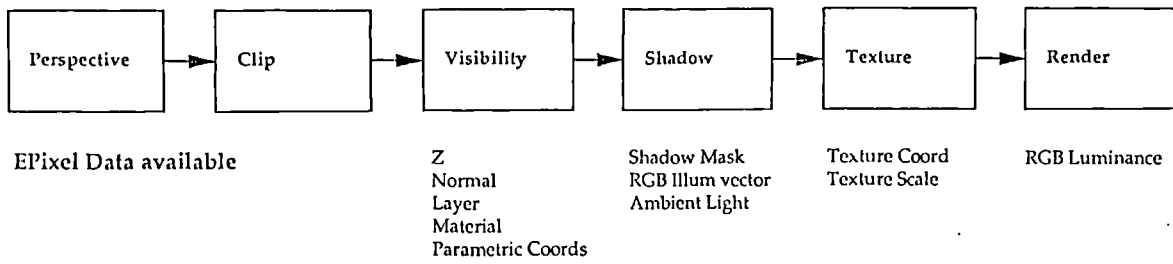


Fig. 23 The Standard Graphics Pipeline

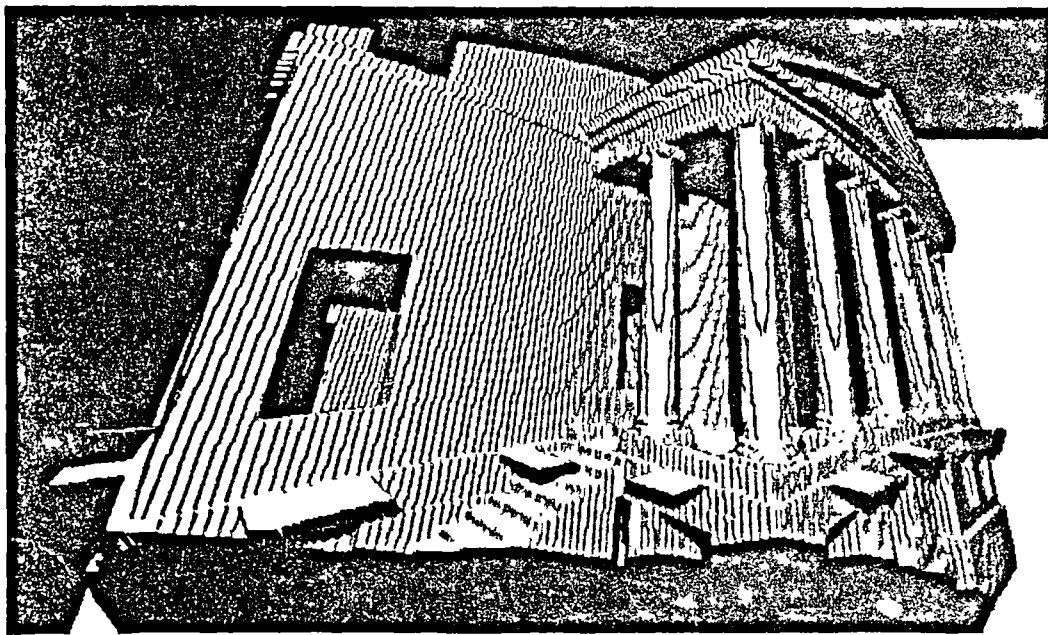


Fig. 24  
A polygon mesh of 3-D points recalculated from an EPixel Image

The most immediate solution to anti-aliasing is to work on images that are large enough to minimise the relative presence of local artefacts. In turn the file size and CPU memory allocation necessary to accommodate large rasters of deep pixels becomes an issue for concern. The standard case scenario is calculated thus:

Header file (perspective projection and lights) estimated at 1 Kilobyte	
RGBA	4 bytes per pixel
Z	4 bytes per pixel
Material	At least 1 byte per pixel for the most basic implementation
Shadow Mask	1 Bit per light per pixel
<i>Total per pixel</i>	<i>80 bits</i>

In this example a PAL resolution uncompressed EPixel image file(768 by 576 pixels) would be approaching 5 megabytes. If the system is to produce high-resolution image then images 4000 by 4000 pixels square must not be out of the question<sup>8</sup>. This is too weighty for many systems; various compression techniques could be used to reduce the file size and run-time decompression could reduce CPU memory usage. These options include run-length encoding, which would be particularly effective on colour, material and tentativeness fields and compression of standard types into binary. Compression techniques should be the subject of further investigation.

It is not expected that all the fields listed above would necessarily be present in an EPixel image. Most successful experiments to date have used just pre-rendered RGB, Z Depth and Material tag. The EPixel image is configured in such a way as to give sensible answers to queries on fields that it does not contain. For instance, if an image does not have any Z data and Z data is requested the image returns 1, which is safe for all calculations.

Another, more radical solution to reducing run-time memory usage, is to emulate the EPixel data type "on the fly" using a ray-tracing port into a scene, somewhat in the manner of Haeberli [90]. This would be more accurate and able to use super-sampling and adaptive sampling techniques which would avoid many aliasing problems [Whitted 80]. The interface to the raster image and the ray-tracing port would be identical. Parallel-processing could be used to increase the efficiency of this approach. However if a scene is to be repeatedly and experimentally re-rendered then the trade-off between the processing required and memory available makes the pre-calculated EPixel image more attractive.

---

<sup>8</sup> The resolution of current 35mm slide imaging devices. An uncompressed image of this size would be 160 Mbytes.

Further compatibility problems between different ports would be resolved by making the EPixel interface resolution independent. EPixel Ports and EPixel Images may be accessed parametrically with point enquiries being made in the parameter space 0-1 in both X and Y dimensions. This would have the effect of separating the output image resolution from the input image resolution. Various interpolation schema would be used to adjust between input and output resolutions which may have adverse implications for edge finding techniques.

## 9.2 Geometrical Methods using EPixel images

The Z channel in conjunction with the a small amount of information on the perspective projection is used for calculating all geometric information from the EPixel image. *The least information* required about the viewing transform is the angle of view in either X or Y. The Z value stored in the buffer is the "real Z", i.e. the distance from any point on the surface in the scene to a plane at the eye point which is parallel to the screen ; the units are configured to represent metres in object space, although this is quite arbitrary. The data available then is, screen X and screen Y location, world Z into scene, the angle of view and the dimensions of the screen in pixels. From this can be derived all 3-Space geometric calculations.

### Calibrating X,Y and Z

If the inverse perspective matrix is available then world X, and Y co-ordinates are calculated accordingly. However, if it is not then the following method can be used to estimate world X and Y, following the standard assumptions outlined in the previous section.

All units in X and Y are in screen pixel steps, and Z is in metres. It is necessary to find the distance into the scene at which a pixel sized step in X or Y represents a 1 metre world-space step so as to calibrate the 3 dimensions into the same units.

```
calibratingDistance = screenWidth / ( 2 * tan( angleOfViewInX ) );
```

### Converting between World Space locations and Screen Space X and Y

Finding the world space co-ordinate of the surface seen at screen X, screen Y is necessary for all world space geometric calculations



```

scaleFactor = worldZ / calibratingDistance ;
worldX = (screenX - screenWidth/2) * scaleFactor ;
worldY = (screenY - screenHeight/2) * scaleFactor ;

```

The converse function which takes a world co-ordinate and returns the nearest screen co-ordinate is also useful

```

screenX = (worldX * calibratingDistance) / worldZ;
screenY = (worldY * calibratingDistance) / worldZ;

```

Figure 24 shows a rendering of a 400 by 600 polygon mesh which has been calculated from an EPixel image. The mesh has been turned slightly side on to the viewer to demonstrate the relief nature of EPixel geometry.

### Calculating normals

Many of the following calculations use a 3 by 3 sampling kernel in order to perform relatively simple differential calculations. A 3 by 3 kernel can be represented thus:-

	top	
left	cen	right
	bot	

Larger sampling kernels may be required for more complex calculation but with consequent reduction in speed. To calculate the normal of point Cen then:-

```

Vector3D tanX = worldSpace(right) - worldSpace(left);
Vector3D tanY = worldSpace(top) - worldSpace(bot);

// cross product of tanX and tanY
Vector3D crossProduct= tanX x tanY;
Vector3D normal = normalise( crossProduct);

```

### Calculating Gradients from the Colour Channel

Certain improvisations can be implemented where a Z channel is not available. The luminance values calculated from the colour channel can be used to infer the gradients of curved surfaces where

the intensity of lighting changes smoothly over the same surface. Shadows and textures cause this method to fail. A simple function using a 5 by 5 kernel is able to return the 2-D Vector representing the direction of most rapid change in luminance, the normal of which is the contour of similar colour [Figures 25a and 25b].

```
changeInX = (farRight/4 + right) - (farLeft/4 + left);
changeInY = (farTop/4 + top) - (farBot/4 + bot);
Vector2D colourContour( -changeInY, changeInX);
```

When no gradient is found (i.e. on flat objects) the function usually returns a horizontal vector.

### Calculating Gradients from the Z Channel

Precisely the same method as above can be applied to the Z channel to find contours of consistent Z, or rates of most-rapid change. This is a far superior technique to using the luminance values of scene, not only is the Z method not prone to confusion due to shadows etc., it is also not limited to curved surfaces as Z usually changes constantly over a flat surface (unless it is face-on; an exception which can be catered for). This technique generates non-perspectival hatching directions that indicate surface orientation [Figure 26].

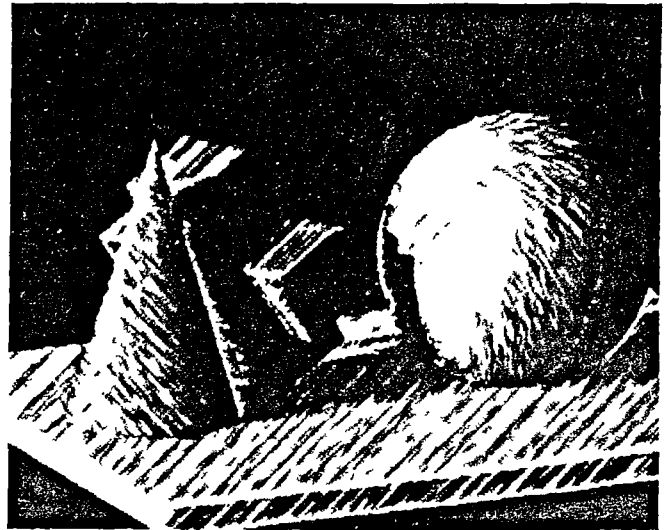
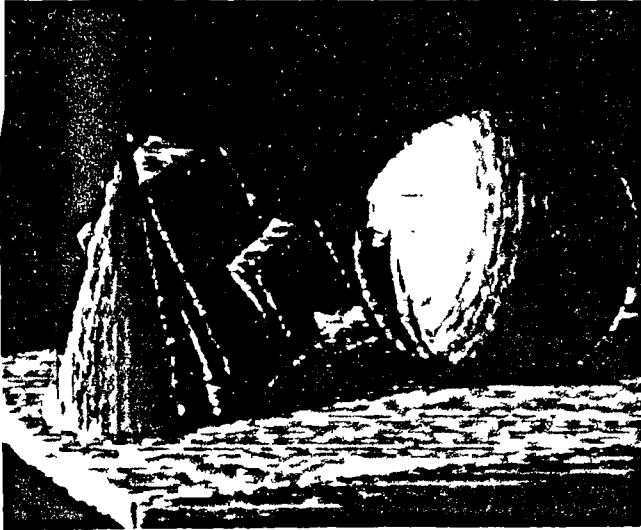
### Calculating screen-based vanishing directions for surfaces

#### A new algorithm

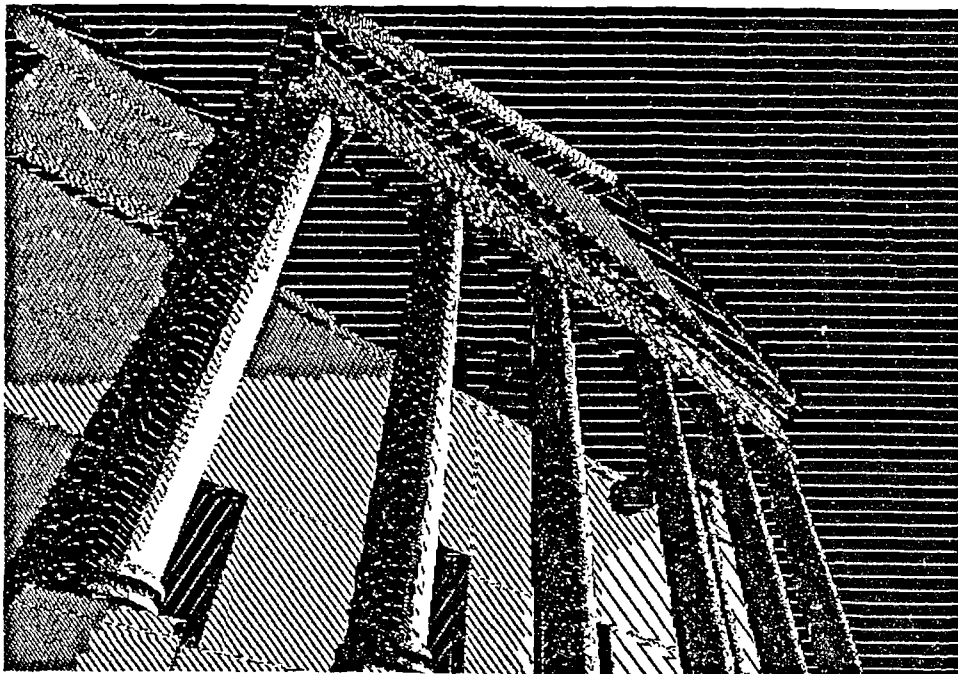
Vanishing points are useful for perspective effects. World-space parallel lines lying on a plane share a common vanishing point, a world-space vector defining the direction of these lines gives a specific vanishing point in the scene for any flat surface. Assuming that the origin of the screen-coordinate system is the centre of the screen and that screen y is "up" in the scene - the vanishing point on the horizon line (world and screen y = 0) is perhaps the most useful but will not work for a surface parallel with the plane world Y = 0. For this surface another vanishing point must be found.

To find the screen-based vanishing direction towards the horizon (y=0) for any point, to give the impression of parallel horizontal lines in world-space, it is first necessary to find the world vector towards the horizon at the corresponding world point.

```
Vector3D toHorizon = surfaceNormal( screenX, screenY ) x
```



*Fig. 25a and 25b*  
*Mark orientation with respect to luminance. The first image orients the marks to follow contours of similar luminance, the second skews the mark by 45°*



*Fig. 26 Non-perspectival Hatching using the Z-Buffer to determine orientation*

```
Vector3D (0,1,0);
```

toHorizon represents a displacement from the original point of inquiry in world space, which when projected back into screen space gives us vanishing direction in 2-D towards screen Y = 0.

```
Point3D worldDisplace = screenToWorld( screenX, screenY ) + toHorizon;
```

```
Vector2D vanishingDir = worldToScreen( worldDisplace ) -  
    Point2D(screenX, screenY );
```

### **A new algorithm for finding edges, corners and curves from Z**

Z values change linearly, within world-space, over a flat surface, and 1/world Z changes linearly per pixel-step within screen-space. Corners, curves and edges can be located by looking for discontinuities in rate of change of 1/world Z within the EPixel image.

```
changeToTheLeft = 1/getZ(cen) - 1/getZ(left);  
changeToTheRight = 1/getZ(right) - 1/getZ(cen);  
discontinuityInX = changeToTheLeft - changeToTheRight;  
Vector2D magnitudeInX( discontinuityInX , 0 );
```

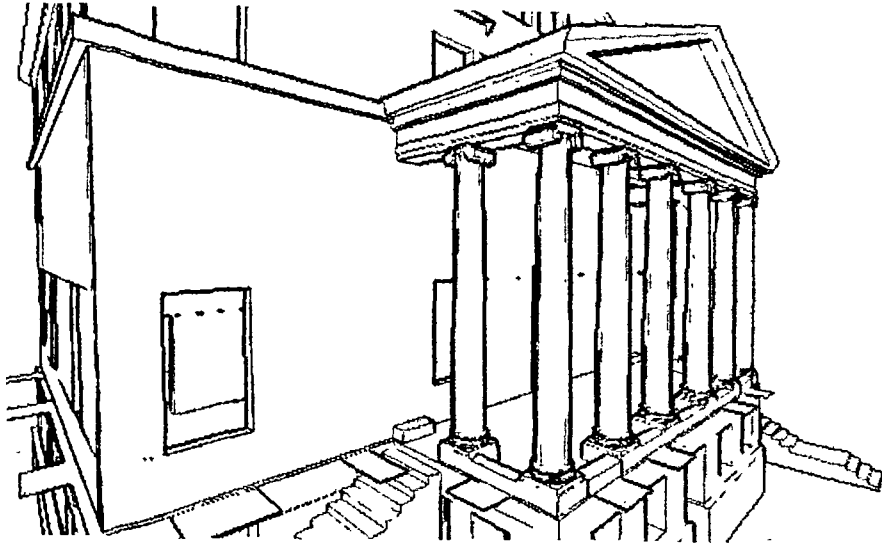
The same is done for the Y dimension. One must be careful to observe the signs of the results. The final result, which is a scalar value representing the strength of discontinuity, is calculated by adding the vector in x with that in y, and measuring the magnitude of the resultant vector:-

```
edgeAcuity = sqrt (discontinuityInX * discontinuityInX +  
    discontinuityInY * discontinuityInY );
```

The range of results from this technique vary enormously depending on whether the point in question lies on an internal corner or a silhouette edge. In the latter case the discontinuity may be huge. Cubing or fourth-rooting the result may be useful for compressing the edge-acuities found in an image into a useful range [Figure 27].

### **Finding directions of Edges**

The precise direction of an edge cannot be found using a small sampling kernel, larger ones may



*Fig. 27 Edges found by finding the 2nd differential in the Z-Buffer*



*Fig. 28  
Ordered dithering used to create tonal variation within an image*

return more accurate results by comparing discontinuities in  $z$  over more directions. One proposed technique is to have a line following algorithm similar to the "find best direction" method used for brush marks in the last chapter.

### Calculating principal curvatures

One of the uses of hatching within conventional drawing is to define the axis of principal curvature for warped surfaces. This is the direction of greatest rate of change. A cylinder has its principal curvature running in stripes along its length, as does a cone. A saddle and torus both have two principal curvatures running at right angles to one another. The sphere and flat plane are exceptions in that the former has an infinity of principal curvatures while the other has none.

Ideally, principal curvature could be found by a variation of the edge direction finding technique. This is an area for future work.

### Interactive photo realistic re-rendering

The source data for the final representation ought not necessarily rely on a pre-rendered version of the scene generated by the modelling/viewing system for final lighting calculations. Instead an unlit scene can be provided. Material types in the EPixel image may contain pointers to further data appertaining to material optical properties. This may be stored as a look-up-table to a finite number of material types as the data required per-pixel would result in huge files (specula, ambient, diffuse reflection as RGB triplets and perhaps emissivity would need to be stored, i.e. another 36 bytes per pixel). When these are used in conjunction with the scene lighting data and the surface normal calculations the scene can be re-rendered using photo-realistic techniques. The inclusion of shadow masks within the EPixel data enables the rendering of shadows. This technique allows for an amount of flexibility over the appearance of the rendered scene. The optical properties of lights in the scene can be altered ; they can be turned on and off or their colour and intensity can be changed. Spatial noise can be added to the world  $Z$  of the scene giving rise to bump-mapping shading effects. Certain techniques are precluded, as this approach is limited to the basic Phong/Gouraud shading model, such as ray-tracing style reflection, refraction and transparency and radiosity techniques.

There may be limited scope for altering the material's optical properties interactively. Either the properties pointed to by the Material Attribute LUT can be changed, giving rise to global changes in the image, or new types of materials can be generated on the fly depending on available space left

in the LUT. This is an area for future work.

### Interface to the EPixel Image

In the final implementation of the EPixel-based renderer the interface to the EPixel data is via a port that can calculate many of the above computations. The current implementation is interfaced thus:-

```
// standard misc
// _____
int xSize();
int ySize();

EPixLayer getLayer(const Point2D p);
EPixMaterial getMaterial(const Point2D p);
Color getRenderedCol(const Point2D p);
float getRenderedLum(const Point2D p);

// screen space info
// _____
Vec2D getContour2D(const Point2D p);
Vec2D getVanishing2D(const Point2D p);
Vec2D getPrinCurve2D(const Point2D p);
Point2D getPoint2D(const Point3D p);
float getEdgeAccuity(const Point2D p);

// screen to world space info
// _____
float getZ(const Point2D p);
Point3D getPoint3D(const Point2D p);
Vec3D getNormal3D(const Point2D p);
```

# Chapter 10

## EPixel Based Advanced NPR Techniques

### 10.1 Point distribution and Line orientation techniques

It is seen from the Expressive Marks system that some effort went into the definition of the distribution of marks over a surface. Generally a density factor was applied to the filling algorithm and the marks bunched-up accordingly, the direction of marks was set overall per polygon. Much more sophisticated distribution and direction methods are required if the images are to acquire the sort of sophistication seen, for instance, in drawings and etchings. A preliminary investigation of this area, with reference to digital techniques, is conducted by W.M. Irvin [1969] dealing with "the micro-structure of various print media – conversion for depicting tone, curvature and texture by means of dot-line patterns".

The last section proposed some techniques for calculating 2-D vectors with respect to surface orientation, perspectives and luminance-gradients. These techniques are useful for orienting marks on the surface; for instance, surfaces of different normals can easily be represented by marks in different directions. Marks following the contours of colour gradients are possible as are marks which converge towards a vanishing point.

In general, where interaction does not happen, the preferred method of distributing marks over a patch of the image is to use a "digital dissolve" technique, which guarantees a pseudo-random visitation to every pixel in the region with very even coverage [Morton 90]. Because the marks are usually much larger than a single pixel the entire dissolve cycle is not run through, usually only a fraction of the cycle is used, this is governed either interactively or via a pre-set maximum percentage.

#### Constraining Point Marks

To achieve varying densities of marks these distributions are varied with respect to some continuously changing data from the EPixel image. In the simplest instance, the distribution of marks might be in relation to the luminance of the surface represented. With stipples and dots a simple dithering approach can be used to thin-out marks in brighter parts of the scene. An ordered dither matrix with 64 scales of half-toning proves adequate for most effects [Holladay 80]. A point is visited in the image via the dissolve effect and the dither matrix (of size  $N$  by  $N$ ) enquired at the modular  $N$  of screen  $X$  and  $Y$  to see if a mark can be placed or not. This is done by comparing the



found matrix value to the scalar value of, for instance, the luminance at the screen X and Y to return a Boolean. The normal artefacts associated with this type of dithering are not a problem as they get fuzzed out by the larger-than-pixel mark [Figure 28].

### **Constraining Linear Marks**

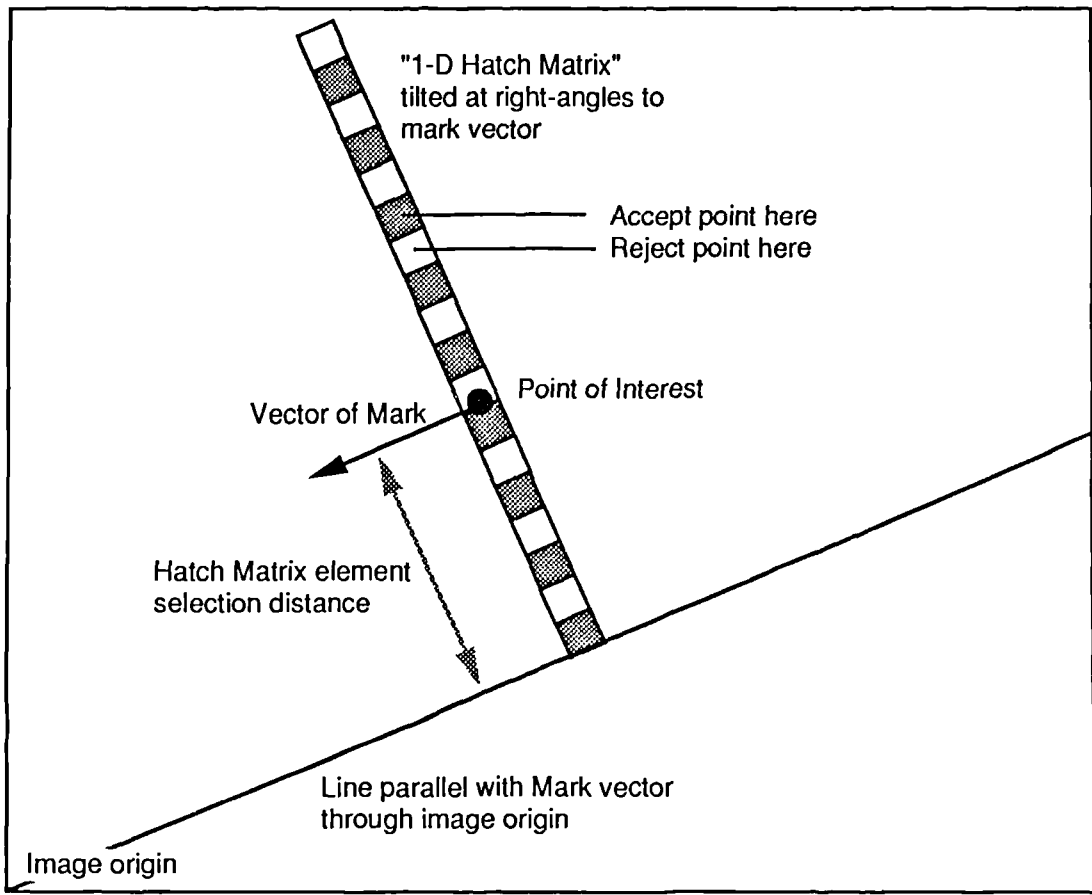
#### **A new algorithm for parallel hatching**

While this method can be applied to controlling the distribution of linear marks the resulting visual densities are difficult to control because the technique does not take into account changes in orientation and length; long marks would make for a denser image than short marks. It would be desirable to constrain their placement with respect to their orientation and length so that predictable densities of linear marks can be formed. This is achieved by constraining marks in such a way as to line them up "head to tail"; the issue of length is therefore partially factored out as the marks run into each other.

A new dither algorithm is proposed which takes as its input a point on the image and a 2D vector, representing the direction of lines or marks required. The algorithm uses an array of Boolean values describing the cross-section of the desired hatch pattern. This is effectively oriented at right-angles to the direction of the mark. The relevant element of the array is calculated by taking the perpendicular distance (modulo length of array) between the mark and the image origin. Figure 29 outlines the basic elements of the algorithm.

Thirty-two levels of regular spaced lines can be achieved by swapping in differently configured hatch matrices for different densities of tone. This technique is effective for generating regular bands of hatching at any angle in the scene. The normal function of this type of half-toning is to return a Boolean for any point in the image determining whether that point can be written or not. When the half-toning matrix and the image pixels are absolutely "square-on" rounding errors do not occur. However, in the case of the tilted 1-D matrix, rounding errors may occur in the calculated position of the query point in the matrix - it may fall somewhere between two elements. Again this does not matter too much with large marks as errors are fuzzed out, but when writing pixel sized dots using this technique jaggies appear in the resulting hatching. A way round this is to return an "acceptance value" as a floating point - i.e. how much "yes" minus how much "no" – and fade the dot accordingly to anti-alias the result. This method is shown used in Figures 25a and 25b in conjunction with the automatic mark orientation method.

The formula for calculating the correct element in the hatch-array of size  $S$ , given a point  $X,Y$  and



*Fig. 29 The Linear mark hatching process*

a normalised vector  $V$ , is this:-

```
element = (( X * V.Ycomponent) - (Y * V.Xcomponent)) mod S;
```

This technique cannot be used to calculate convergent lines, for perspectival effects, as a surface whose hatching lines are tending towards converging around the origin of the scene cause the algorithm to fail – it cannot find the correct element of the hatch-array as all measurements resolve to zero.

This approach has been since modified to calculate hatching using a simple clipped sine function instead of an array.

### **Perspectival Hatching**

#### **A new algorithm for Perspective Hatching**

Perspectival hatching is a particularly desirable effects for a non-photo realistic renderer to have; scene surface geometries can be delineated very elegantly using this technique. One required effect is that the hatch lines converge towards the vanishing point of the surface; as they do so they thin-out to avoid excessive bunching. The hatch lines themselves are of even thickness over the image. A technique is proposed for generating such effects. The user must specify the actual metre-spacing of the hatching in the scene at a certain distance into the scene<sup>9</sup>.

Firstly the hatch spacing in meters must be set for a specific  $Z$  within the scene. The thinning algorithm cannot work continuously as this would debar any continuous hatch lines from forming. Instead the lines are thinned at regular intervals of  $Z$ .

For any world point  $P$  within the scene, with desired spacing at  $M$  meters of  $S$ , thinned at intervals of  $N$  meters

```
roundedZ = (int)( worldZ_at_P * N )/N;  
unthinnedSpacing = S / calibratingDistance; // see page 82  
thinnedSpacing = unthinnedSpacing * (roundedZ + M);
```

---

<sup>9</sup> The hatching deemed useful for architectural renderings was to specify a spacing of 0.2 metres at 10 metres into the scene and thin out every 5 metres of  $Z$ , although smaller distances and closer hatches may be desired for smaller-scale scenes.

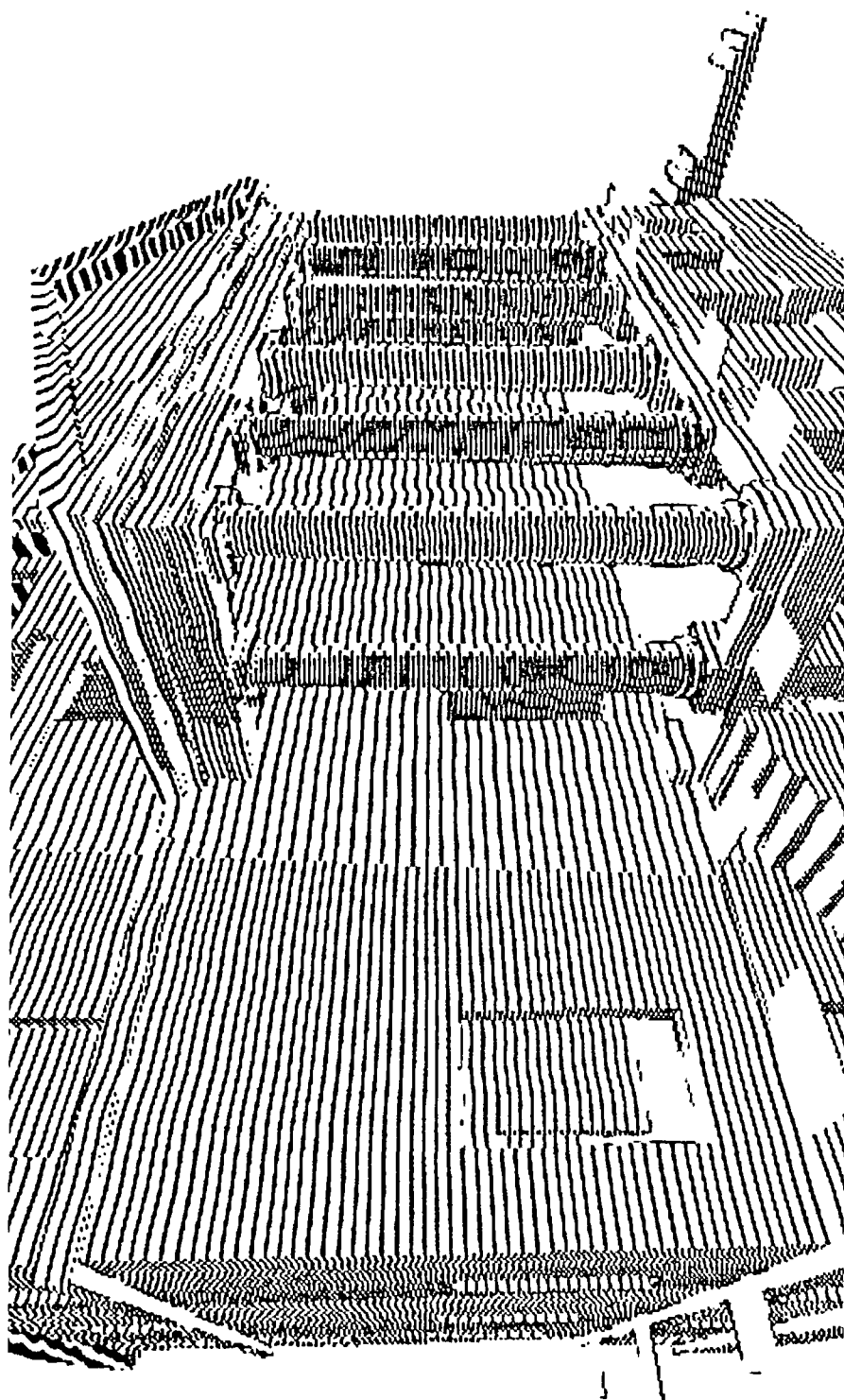


Fig. 30 *Perspective Hatching*

Once the world space metre spacing has been calculated at point P a spatial constraint algorithm checks to see whether point P is accepted or rejected. This is done by calculating the world point at screen point P and then taking a modulo (modulo width of hatch) of the world Y component of this point. This is compared it to the desired hatch spacing; if it falls within set limits the point is accepted. These calculations occur in world space and the resulting hatching can therefore be thought as a type of solid texturing [Peachy 85]. However, to avoid the bands of accepted points getting broader and narrower within the scene they have to be constrained in two ways; firstly with respect to world Z, and secondly with respect to the orientation of the surface. Where the surface has a shallow gradient, without further constraints, the hatch would become broader as the Y dimension is cycled through more slowly over such surfaces.

```
slopeConstraint = 1 - surfaceNormal(P).normalised_Ycomponent;

modularY = abs ( remainder( worldY/thinnedSpacing ) );

if ( modularY < (worldZ/calibratingDistance)*slopeConstraint )
    return TRUE;
else return FALSE;
```

Artefacts generated at the thinning-step interval can be fuzzed out by introducing 2-D spatial noise to the effect. This can also be used to wobble the hatch lines in Y to create a more interesting texture. Perspective hatching is used in conjunction with the vanishing point vector algorithm to place marks within the scene. These effects are shown used in Figure 30.

Other perspectival effects, such as parallel narrowing lines ( the "sleepers" of tram-tracks rather than the "rails") may be calculated using similar methods

## 10.2 The Drawing of more Complex Entities

The techniques discussed so far are ways of representing surfaces and materials in the scene which, while not being photorealistic, are never the less wholly dependent on the luminance, geometry and material properties of the model.

A very powerful supplement to this range of techniques is the inclusion of reasonably high-level,

explicit descriptions of more complex elements such as leaves, bricks, clouds, complex textures, grass and even people. These are then applied in various ways to add detail to the output image, where no detail actually existed in the EPixel image.

Two methods are examined for explicitly defining more complex entities; painted bit-maps and drawn display-lists. Both can be applied to the image directly or using various perspectival and stochastic transformations.

### **Bit-map defined shapes**

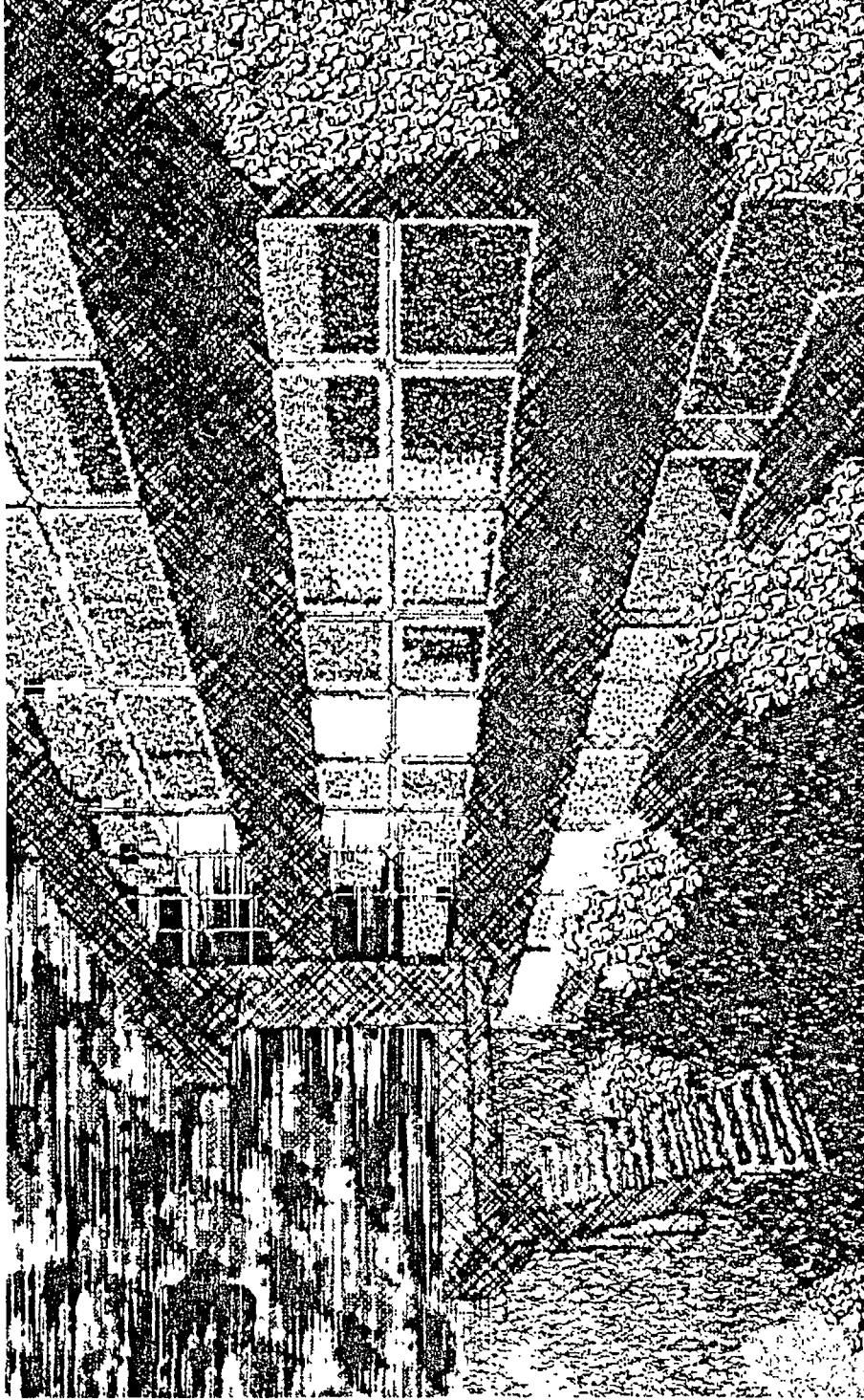
These can be defined using any standard paint system and constitute an alpha mask. This can be used for block-filling areas, and the use of edge-finding techniques can render their outlines, either as another extracted bit-map or as a path in the form of an auto-traced set of vectors given to a linear mark for painting. The leaves and clouds in Figure 31 have been defined using a 1-bit deep bit map, thus no soft edging has been used, as would be possible with a deeper bit-map. Obviously there is a trade-off between bit-depth and speed in such a system.

The bit-maps used in Figure 31 have not been transformed in any way (other than shifting and clipping). However an area of great interest for future research is in the perspectival and stochastic transformation of these cut-outs. The perspectival transformation of bit-mapped images into a scene is similar in many respects to texture mapping, but would allow greater freedom, over-layering and consequently greater visual variance than the former. For this technique the perspective transform is calculated for a single point within the scene that represents, for instance, the centre of the cut-outs final position. A new transformed cut-out is calculated and applied. Hence, cut-outs in the final image may overlap. Clipping against found changes in surface normal will add clarity to the image and avoid the obfuscation of edges and corners. It is suggested that the transform applied to the cut-out does not necessarily have to be 100% accurate to that calculated for a specific point. Instead it can be a blend between no-transform and full transform, achieving slightly unrealistic effects of the sort seen in the paintings of Henri Rousseau .

Stochastic effects may be used to perturb the final transformed cut-out to avoid visual repetition within a scene. Noise filters and blotter techniques can be used to erode or grow the shape in random ways prior to perspective projection.

### **Drawing Display-list defined entities**

Figure 31 also shows the use of small display lists applied to the scene to draw grass and the wavy



*Fig. 31 Scene rendered showing the use of more complex graphic elements such as leaves with outlines and grass*

lines in the sky. These are defined using any interactive drawing package and in the case of Figure 31 the grass is transformed on Z to rescale the drawn entity while the "sky-lines" are not scaled. The final implementation will support full perspective transformation and clipping of the sort described for the bit-map cut-out. Generally such techniques are easier for display lists as they are computationally less hungry and are less prone to aliasing problems.

Stochastics perturbation is also easier to apply to display lists than to bit-maps. Fractal techniques have proved particularly successful at producing highly controlled variations such as mid-point displacement.

### **Spatial application**

While a random scattering of these more complex entities may be sufficient for leaves, clouds and rubble (i.e. organic, random configuration), it is not an adequate technique for regular or geometric configurations seen in bricks, tiles or wall-paper. In order to place the elements correctly in relation to one another more precise spatial constraints need to be used. A variation of the perspective hatching algorithm, which generates hatching both horizontally and vertically, then finds points where the two hatches intersect, may be used to place tiles, for instance. This is an area for further work.

### **Depth Sorting and Clipping**

Because many of these larger, more complex entities can run over potentially large parts of the scene, the need for certain types of depth sorted clipping emerges. In the current implementation the entire graphic element, cut-out or display-list, is assumed to be a flat (face on) object whose depth in the scene is the Z of the pixel sized sample point. This can be used for clipping against objects in the scene. In the case of Figure 31, grass can be show growing in front of the building, but is clipped against the underside of the stairs. This is achieved by clipping only when a *different* material type is found to be *in front* of the grass.



# Chapter 11

## An object oriented approach to defining and applying NPR techniques

We are now in a position to consider how this functionality of the mark-making, mark-controlling (distribution, orientation, clipping), geometric calculations and other drawing decisions might be encompassed within a single frame-work. This frame work must facilitate the authorship of rendering styles in ways that are intuitive, flexible and extensible. An object-oriented system is proposed in which the user builds drawing technique or "Sketching" descriptions as a library of functions. These can then be used in the rendering of a scene through interactive or automatic application. I have called this part of the system the Sketcher system<sup>10</sup>, which is a tool kit for designing and authoring NPR techniques – it forms the 2nd important part of the completed Piranesi System.

The Sketcher system grew from looking at certain types of drawings and considers the action of drawing to be fundamentally a hierarchy of nested capabilities (see Section 4.5). This suggests a similar hierarchical nesting of ability within the system. "Trees" of embedded algorithms are configured by the user, at each node a set of unique instructions and operations guide the action and technique of drawing. At the bottom of the tree – after a series of decisions have been made, data gathered and instructions followed – a graphic element (mark, texture, dot, effect ) is applied to the image.

### Fundamental Capabilities of the Sketcher System

Several fundamental capabilities have to be catered for, these are :-

- The ability to make marks in the scene using a choice of graphic elements
- The ability to control the placement and style of these marks in response to the incoming EPixel data
- The ability to fire-off new sketching algorithms automatically when certain conditions are met

---

<sup>10</sup> So called to parallel the "Shader" system used in PR rendering systems  
[Upstill 89]

- The ability to split the scene up into discrete regions, to attribute a certain sketching method to a certain part of the scene.
- The ability to build a Sketching algorithm that places the sort of mark desired in the correct place
- methods for automatic control and user interaction

## 11.1 The Sketcher Tree

Let us now consider a typical Sketcher-tree for rendering a scene composed of just two separate elements –ground and sky. We wish the ground to be rendered as drawn grass clumps, which fade and get smaller with distance, and the sky to be a stylised cloud-scape composed of scribbles and patches of texture.

Two sketching sub-trees are loaded into a Top Level node - a "Grass Sketcher" and a "Sky Sketcher" – from a library of pre-configured Sketchers. These are attributed to their respective material types (ground and sky), perhaps interactively, or they may already have their attribution set. The attribution is stored in the upper-most node of the specific Sketcher sub-tree, which is always a Predicate Node.

A point is selected in the scene, either by hand (mouse interaction) or automatically via a point picking function, and this is passed to the top level node. The top level node queries each of its child nodes to see if any of them is a Sketcher for that particular type of material. The Predicate Node tests the incoming data to determine whether or not the point in question is to be rendered. If no Predicate Node is included then the task of rendering a point will always be accepted. If the point is accepted, the Top Level Node passes down the control and data gathered so far.

All Sketcher nodes contain active functions for passing a bundle of gathered data to their child nodes in a depth-first traversal of the entire tree, starting with the top level node. As the bundle is passed down the tree, through various types of nodes, more data is added or manipulated. At the end of a series of nodes it is assumed that some actual drawing is done to the image. Apart from these "end" or drawing nodes (which contain no children), all nodes can contain as children any other type of node in any order. The Top Level node is also an exception – it cannot be the child of another node. In this way a type of "drawing hierarchy" is defined which can be notated either

visually (using tree-diagrams) or symbolically.

Considering the grass-sketching branch of the tree, once a point has been passed to it, it is passed down to a Drawing Script Node, which contains a cardinal description of a clump of grass as a small display list. Using the geometric information available from the EPixel image, coupled with its own stochastic perturbation and perspective transform functions, a transformed copy of the drawing is generated and added to the bundle. This is then passed to the Mark End Node, an end-node, which executes the drawing of the clump with a pre-defined mark. The mark may have been pre-figured to fade with distance. Once the drawing has been completed the control traverses back up the tree to the Top Level node to deal with the next screen point.

If that point happens to be part of the sky then the Grass Sketcher Predicate Node will reject it and it will be accepted by the Sky Sketcher tree. The Sky Sketcher splits into two sub-trees via a Switch Node. One sub-tree is used for the placement of bit-mapped cut-out shapes in the form of wavy translucent clouds using a Bit Map Node and a Texture End Node. The other is for drawing horizontal lines using a Drawing Script Sketcher and a Mark End Node. These two sub-branches are in turn predicated by Predicate Nodes, determining the frequency (probability) of one effect versus the other.

While only a few different nodes are described here many types of nodes are envisaged in the final system enabling much more complex decision making and data modification processes to be built into any sketching operation. These nodes can be subclassed to perform more specific operations.

Figure 32 shows a schematic representation of the 3-Sketcher tree. Described above are the Grass and Sky Sketchers while the Block Sketcher is a simple stipple added to a spatial constraint (Ordered Dither) node. Figure 33 shows the image produced using this configuration.

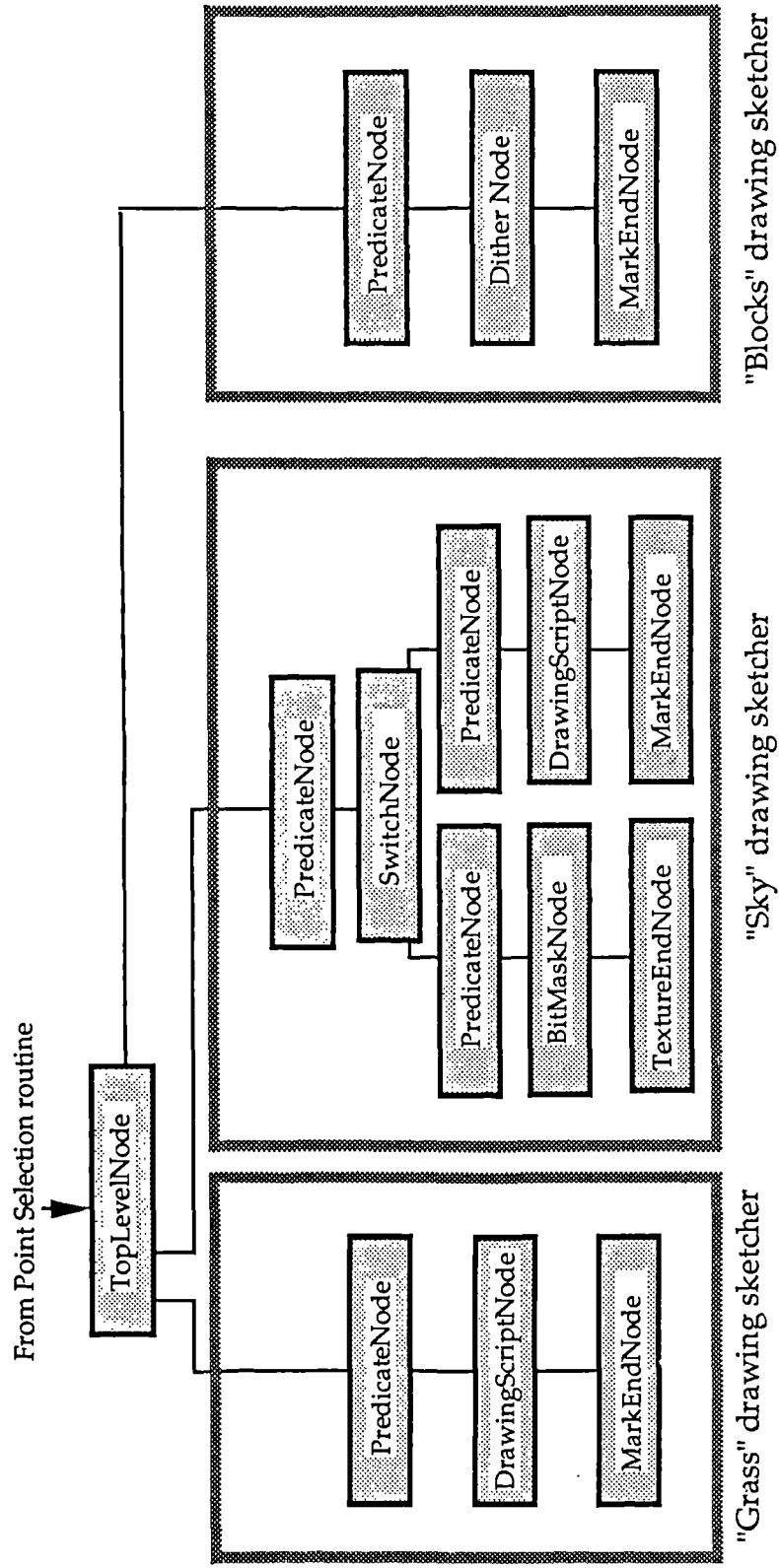
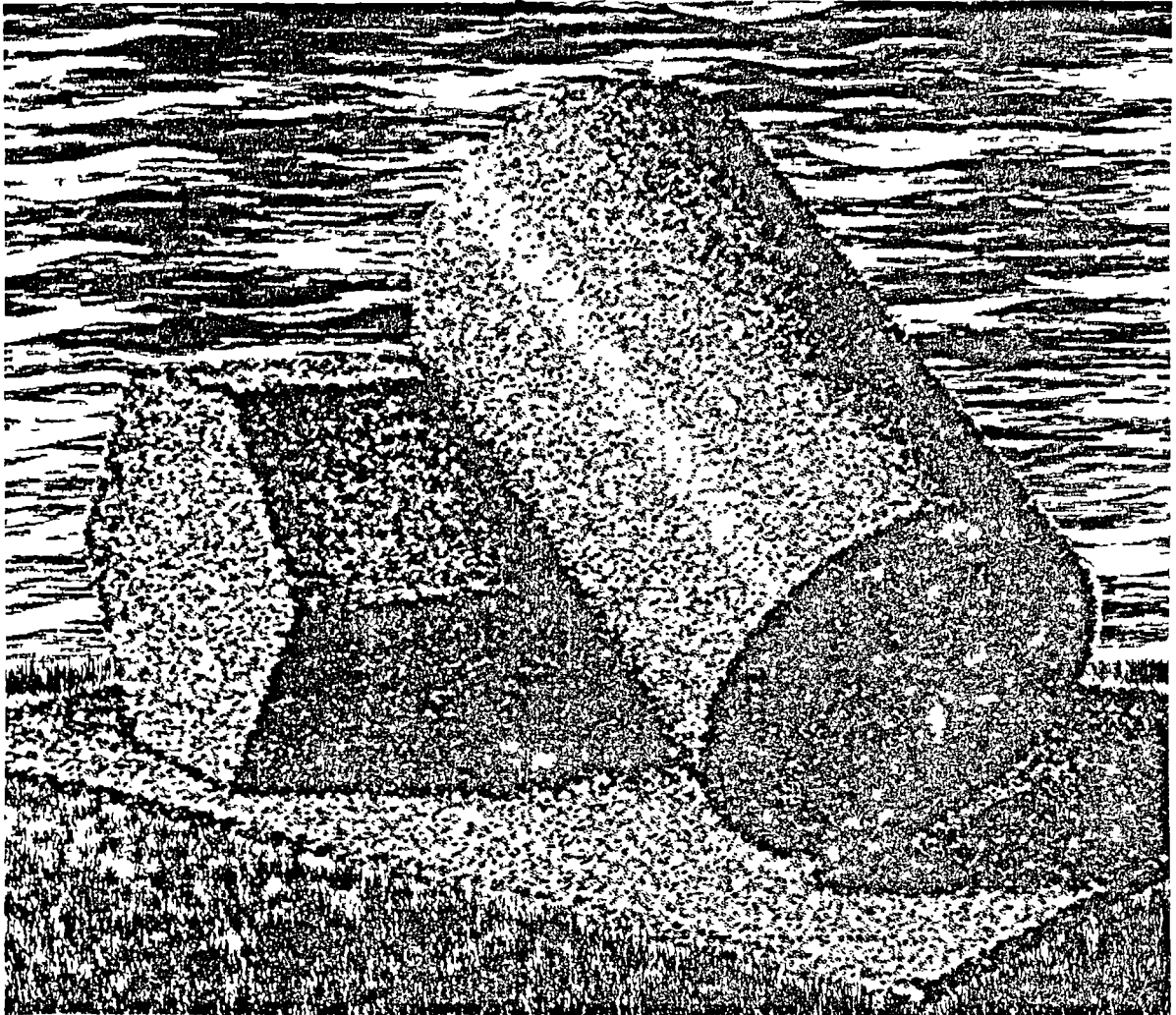


Fig. 32 Assembly of Nodes for rendering a simple scene  
see Figure 33 for end results



*Fig. 33 Scene rendered with a 3 separate Sketchers*

## 11.2 The Control Flow Mechanism

There are three basic abstract types of node from which all other nodes are derived. They are the Thru Node, which has one parent and one child node, the Switch Node, which has one parent and several children, and the End Node, which has one parent and no children and is generally used to write to the image. An exception to this is the Top Level node, which has no parent node.

Nodes pass control to their children during render-time using the function `getControl`, which has in turn been called by their parent node. The whole Sketcher tree, during run time, can be thought of as many `getControl` functions nested within each other. Within a Thru Node the function looks like this

```
void getControl (DataBundle dataIn)
{
    if( controlIsNotOn() )
    {
        modifiedBundle = modifyInfo( dataIn );
    }
    turnControlOn()

    myChild->getControl( modifiedBundle );
    if ( myChild->controlIsOn( ) ) return;

    turnControlOff()
}
```

Control may linger in a node over several calls, depending on whether or not its child node has finished its business. The data bundle is only modified on the first accepted call to the `getControl` function of any node, and modifications cached for subsequent calls. When the child node has finished its operations (indicated by it losing control), the parent `getControl` function turns its control off and the cache is destroyed.

The `modifyInfo` function is virtual and is subclassed for each node type. Usually the acceptance or rejection of a render-task depends on the inclusion of Predicate Nodes, however other nodes can effect control flow in other ways. The Switch Node `getControl` function has a selection of children to iterate through or can select just one depending on how it is set. A Spatial Constraint Node for instance, on determining that a point is NOT to be rendered, is able to set a Terminate Control flag in the modified data bundle forcing any child node to reject control. The End node replaces the `myChild` calls with specific drawing functions. When all control along a certain branch has been

exhausted the control recurses back up to the tree.

## 11.3 Description of Sketcher Nodes and other Elements

### The Data Bundle

This is an object which is passed between nodes as the Sketcher tree is traversed. At each point in the node, actions occur influenced by incoming data, and further data is added to the bundle. The bundle can contain:-

- 2D Point of Interest
- A Drawing Display List
- A Bit Map or Alpha Mask
- Scalar Intensity Amount
- Mark Direction vector

The Data Bundle can become quite large throughout the traversal of a Sketcher tree and for this reason it is passed as a reference or pointer from parent to child, rather than new copies being made between each connecting node. A certain points in the traversal of a tree modifications need to be made to the bundle; at these points the bundle is copied by the modifying node and the copy is modified and cached. This means that changed bundles only propagate down the tree thereby avoiding the danger of redundant data propagating back up the tree after a series of completed operations.

### Abstract Base Class Node

This defines the functionality common to all nodes. For instance, all nodes have direct access to the EPixel image via a global pointer to the EPixel Image Port. The abstract base-class node is a subclass of the SGI Inventor SoGroup node [Inventor 92]. This means that all nodes have automatic reference counting ensuring correct construction and destruction of themselves and their children throughout the execution of the program. Other functionality inherited is the ability to access parameter data of any node, no matter how deeply nested, throughout run time and the ability to load and save arbitrarily complex Sketchers to and from file.

### **Switch Node**

Constitutes a branching in the Sketcher tree between several possible sketching sub-trees . Usually the child nodes are polled sequentially and control passed down if accepted. However a Switch node can be set to select only one of its children for control passing ; this is done using Stochastics to choose the order of poll.

### **Top level node**

This is derived from the Switch Node and contains the top levels of all Sketcher branches. It has methods for adding new and deleting old Sketcher branches throughout run-time and can be thought of as a store-house for adding sketching functions for the execution of a complete render. Input either from an automatic point selection algorithm or user interaction is passed to its children nodes.

### **Predicate Node**

This is mainly responsible to accepting or rejecting control based on properties found in the EPixel file at the point of interest. The criteria at its most simple is usually the material or layer type found, but they can be quite complex i.e. Material: Building->glass, Scalar Values : Z between 10 metres and 30 metres and Colour Intensity between 0.2 and 0.6 with a Probability Factor of 0.7.

### **Drawing Script Node**

This node contains a drawn description of some complex element as a display list. It is able to transform this drawing according to input geometrical information and user settings to generate a transformed copy which it then adds to the Data Bundle and typically is used by a linear -mark drawing node to write to the final image. It contains stochastic perturbation functions to vary every drawn instance in controlled ways. It also contains thinning algorithms for drawing at certain distances i.e. certain lines are omitted in the transformed version at various values of world Z or when foreshortened through perspective transformations to oblique angles. Drawing can be set to clip against different materials of closer Z and change in normal.

### **Bit-mask node**

This is similar to the Drawing Script Node but contains a bit map image instead of a vector display list. This can be transformed according to received geometrical information and user settings. This is then added to the Data Bundle and typically used by a texture area drawing node to write to the final image. It contains stochastic perturbation functions to vary every instance in controlled ways. A derivation of this is the Alpha Mask Node, which is able to hold some tonal and textural information.



### **Area Selection Node**

This is derived from the Alpha Mask Node, but contains its own methods for generating an alpha mask from the EPixel data based on selection or flood-fill algorithms. It can therefore strip out large chunks of an image, perform some transformation on them, and pass them down for processing, typically for applying large areas of texture as in a background wash. It also contains methods for feathering or strengthening the mask at its edges. To avoid repeated picking of parts of the image the Node can be made to observe the Already Done mask (see later). This is an area for further work.

### **Spatial Constraints Node**

This is used for Dither, Hatch and Perspective Hatch functions to determine whether a given point can be rendered or not, in which case the Terminate Control flag is set in the Data Bundle. If anti-aliasing is set then a scalar intensity is calculated, rather than a Yes or No. It is initialised by a description of preferred Hatch directions and Dither densities per given luminance, although it can be set to take principal curvatures, vanishing directions and colour gradients into consideration. This can be "skewed" by a pre-set amount. A mark direction vector and intensity value calculated and this added to the Data Bundle.

### **Pattern Map Node**

This is derived from Spatial Constraints Node and processes points in the image for acceptance, rejection, or scalar intensity setting to define more elaborate configurations. This is based on multiple perspective hatching calculation to determine intersection between perspective hatches of various sorts. Thinning algorithms are applied to the pattern as it recedes. This can be used in conjunction with Drawing Script Nodes to perform complex, textured drawings, such as brick-coursing, tiles etc. or used with a Alpha Mask Node to generate a sort of stochastic texture map.

### **End Nodes**

These currently are Mark, Dot, Texture, Effect , Photorealism Node (see "Graphic Elements" section 8.5). The generic capability of end nodes is that they have direct reading and writing access to the output image via a global pointer to the output graphics port.

Dot, Mark and Texture nodes have methods for setting the colour of the written graphic element. This can be a simple colour, including opacity (alpha) or it can be via sampling the RGB EPixel image at the Point Of Interest. Colour transforms between the sampled colour and the written colour can be established.

Mark, Texture and Effect Nodes have methods for setting clipping of various sorts. The most basic is to clip against the material region. Clipping against scalar values can be set. This repeatedly checks the EPixel image for discontinuities in the information, such as edges or colour values. In both cases under and over-running of clip boundaries can be set. The Mark Node has methods for finding best directions on the RGB channel.

### **Functional Priorities of the End Nodes**

Taken at face value the Data Bundle can contain data or commands which seem to be contradictory or ambiguous. The Mark Vector value seems to clash with the Drawing Script data, and these in turn clash with the fact that a Mark Node may have the find Best Direction method active *and* a default direction. With most End Nodes the situation is unambiguous because most only consider a point of interest and a scalar value, with no directional information. However, certain End Nodes need to parse incoming information to resolve ambiguities, as in the case of the Mark Node. These conflicts are resolved by default and user set priorities. The default priority order of Data Bundle information for a Mark Node to observe in descending order is, Drawing Script information (greatest priority) , Mark Direction vector (this is derived from geometry information) , internal best direction (if set), default direction (least priority).

### **Blending**

Switch Nodes, Predicate Nodes and Spatial Constrain Nodes are all used as part of an elaborate "decision making" process for drawing techniques as the tree is traversed. This embodies a lot of the functionality associated with drawing as discussed in the analysis of drawing (sections 4.4 - 4.8). However as was also discussed in those sections, there are many effect in NPR that are not decisions to use one device or another, but various blends between devices.

Blending is build into the Sketcher class system at various levels and is used to automatically calculate parameters based on scalar values from the EPixel data. For instance, any two similar End Nodes can be inter-blended with respect to luminance or Z field in the EPixel data; one might be a strong mark, the other a faint mark and the blend set to produce marks between the two, thereby creating faint, distant marks and strong, near marks.

In order to create a blend effect in a specific End Node, it must be given two other nodes representing the extents of the blend. These are stored as children, although not directly used to do drawing. Instead the children's parameter fields (Based on the Inventor SoField system) are used as the extents of a parametric blend based on the EPixel data. The resultant value is stored in the parents own fields at which the newly parametised mark can be drawn to the image.

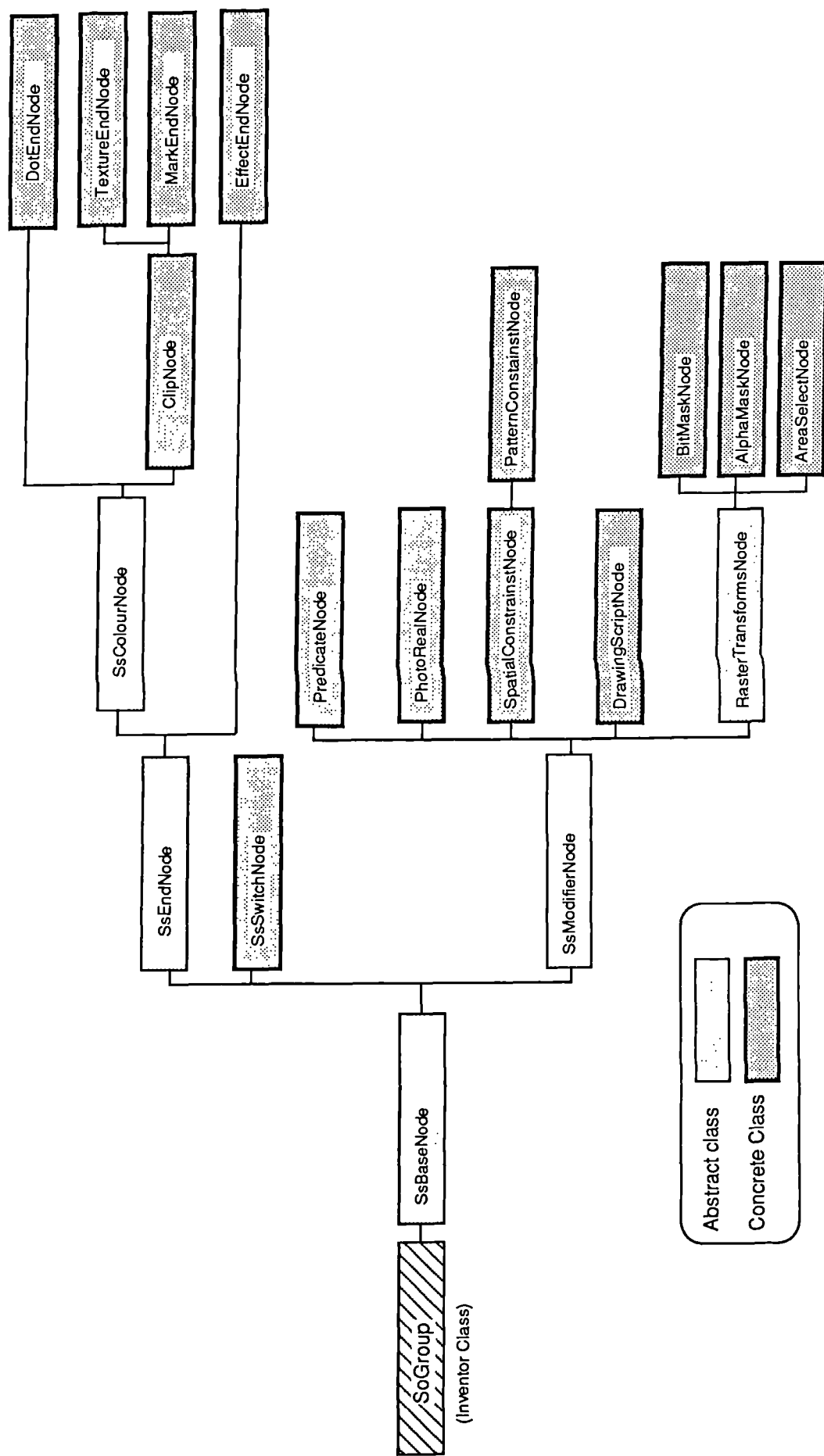


Fig. 34 Inheritance hierarchy for the Piranesi System. This is built on top of the SGI Inventor node class for its data-base and reference counting functionality.

In order to achieve blending an End Node must contain two children and have the `setBlend` function called describing the mode of response to the EPixel Data.

Figure 33 shows the class inheritance hierarchy for a prototype Sketcher system.

### The Already Done Mask

A final set of image writing criteria are calculated by referring to a mask that monitors which parts of the image have been done and by what. This can be used for various screen-space calculations and to avoid re-painting of parts already done. This requires further investigation.

### Loading, Saving and Editing Sketcher trees

This will be done by subclassing the entire Sketcher hierarchy from the Inventor class `SoNode`. All member state variables are stored as registered types which are accessed for modification during render-time or file reading/writing by standard Inventor Actions. The Inventor database contains registered Node Types and Node Fields and is able to save the type of node and its parameters thus:-

```
SwitchNode {
  myPollModeFld    POLLALL
  Hatch2DNode {
    mySpacingRangeFld    0.7
    mySpacingFreqFld     4.4
    myThreshRangeFld     0
    myThreshSetFld 0.47
    DotNode {
      myBlendFlagFld ~
      myBlendLoFld  ~
      myBlendHiFld  ~
      mySampAmountFld    1
      myColVarAmountFld  0
      myColModeFld    OWNCOLOUR
      myColVarModeFld    NO_CV
      myColourFld    0 0 0
      myOpacityFld    1
    }
  }
}
```

Here the Node types have been shown in bold, and the Field types in normal type. Note the nested nature of the description – this is used to re-build the tree correctly upon re-loading.

Render-time modification of parameters can be done in one of two ways; either the entire Sketcher

is deleted and rebuilt using different parameters when calling the Sketcher Library or a run-time editor can be used to access any node within the tree. This latter solution is more desirable and has been shown to work for any potential Inventor node by Tim Wiegand at the Martin Centre, University of Cambridge. It is used in the Model system [Model 93] for editing lights or material types. In the case of the Model system (which is a 3-D modeller) the user asks that a specific node be edited by selecting an object in the scene and hitting the "Edit" menu. The system interrogates the specific node to see what fields it contains and what types they are and constructs an interface window "on the fly" using a range of Motif widgets [Motif 92]. This as yet needs to be integrated within the Sketcher system.

## 11.4 Combining the Elements to Form the Piranesi System

Figure 34 shows an overview of the completed Piranesi system. It integrates the EPixel image concept with the Sketcher system in an interactive framework .

Figure 35 and 36 show prototype interfaces to the system

## 11.5 Summary and Further Research

A reasonably good, robust, extensible structure for a NPR system has been proposed and the major conceptual problems resolved. The Sketcher system combines a versatile method for defining arbitrarily complex rendering styles and several methods for applying them to a 3-D scene, some of which are automatic while other are interactive.

Apart from those areas for further research already indicated in the text other areas need to be addressed before the Piranesi System can meet the demands of the Design community. While the system works at a functional level – i.e. the computability of the problem has been resolved – many of the User Interface issues still need a great deal of attention.

The authoring and tweaking of Sketching algorithms is still very un-intuitive. It requires knowledge of the specific functionality and methods of the Sketcher system which often has very little to do with the familiar notions of painting and drawing. As such a non-computer literate designer would have a great deal of difficulty using the system as it stands. This is partly redressed by the ability to build large libraries of pre-set Sketchers and the ability to save large

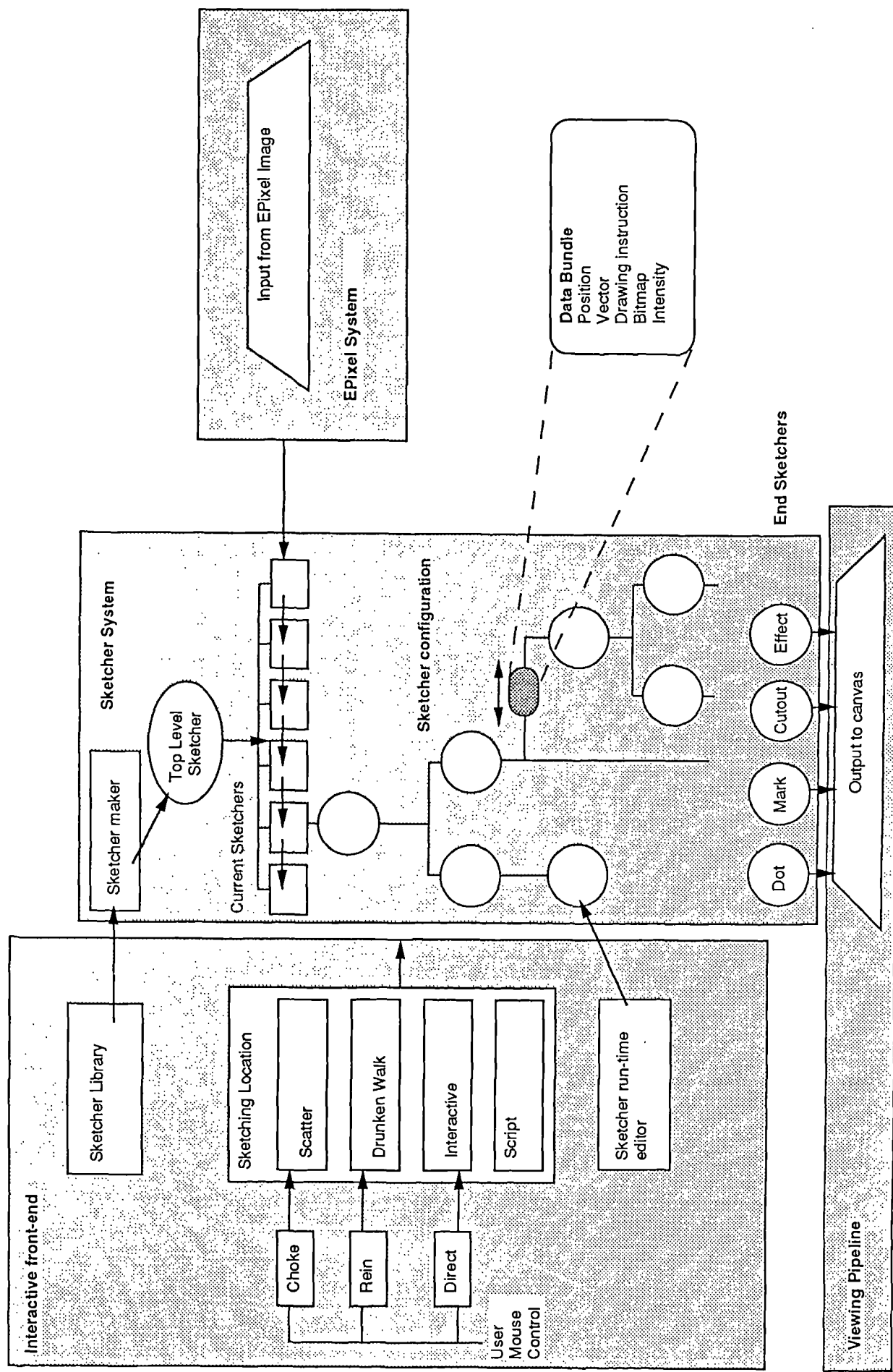


Fig. 35 An overview of the final Piranesi Application

numbers of Sketchers as files. However, their tweaking to achieve variations around the pre-set is still rather hit-and-miss.

An intuitive interface that allows the authoring and tweaking of Sketchers in ways that artists and designers will understand has yet to be built.

The method of image making interaction is still only vaguely understood. This new rendering paradigm of semi-interactivity has only just begun to be explored. Good techniques and strategies for computer/user co-operation rather than computer/user interference and agitation will only emerge with time and use. An important phase in the investigation will be the monitoring of the systems ability and limitations in a proper design environment.

It is intended that along side the novel NPR techniques offered by the Sketcher System will be more conventional painting tools such as Cut and Paste, alpha-masking and area selecting, filters and standard mouse-based tools like paintbrush, eraser and smudger.



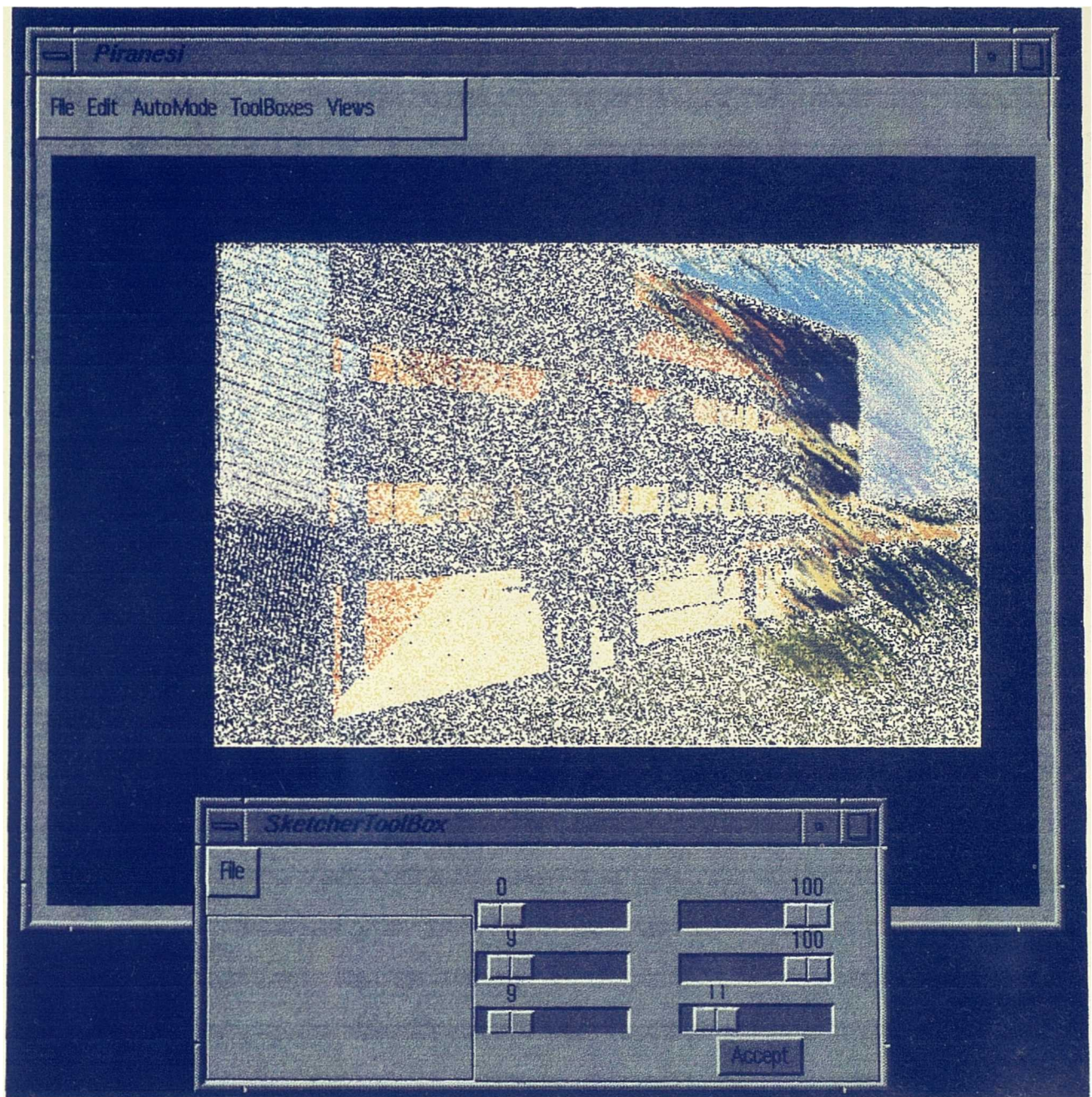
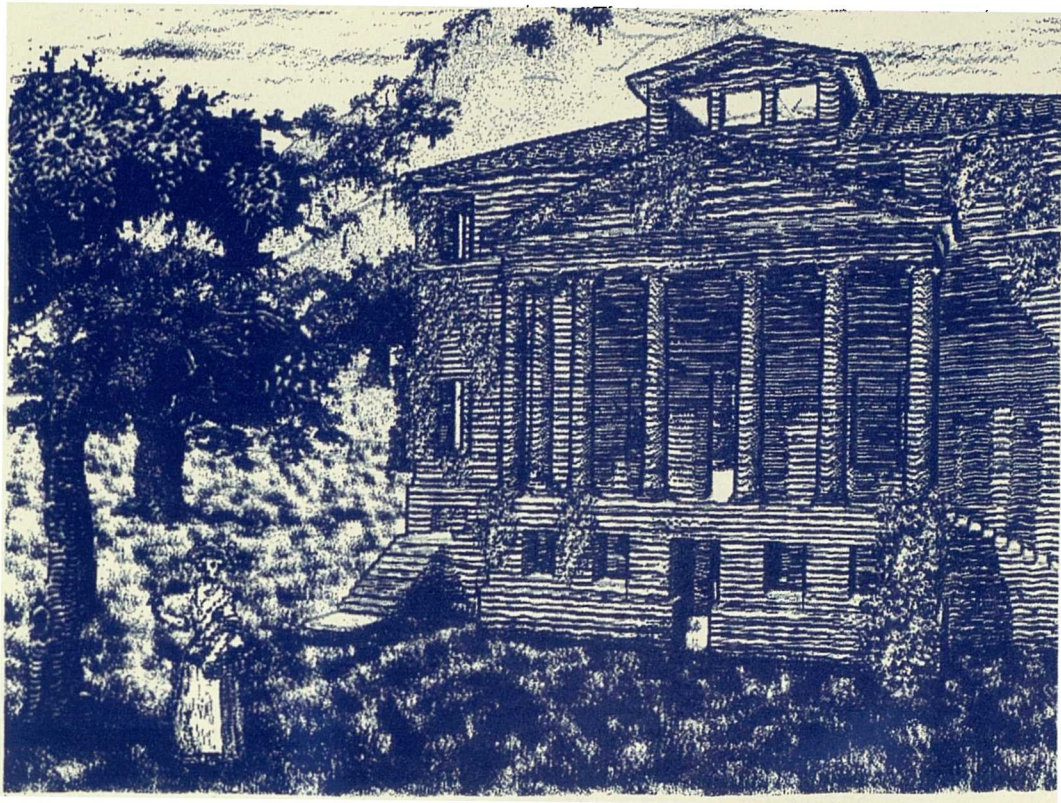


Fig. 36 Interface to the prototype Piranesi system





*Fig. 37*  
*Several Techniques used to create "neo-classical" renderings*



*Fig. 38*  
*Solid Textures overlayed to enhance a photorealistic image*

## Chapter 12

### Conclusions

This thesis began with the claim that, while there are several exemplary NPR systems either published or marketed at present, there are several areas that need to be addressed before NPR can be considered a mature representational tool. Initially, the thesis asked for a better definition of the aims and goals of NPR and further, a better definition of NPR on the whole. This review has been conducted against the current background which sees CG largely dominated by PR imagery. It is largely in response to this dominance that NPR strategies for image making have been developed, either for aesthetic or practical reasons.

The review of current NPR systems in Chapter 3 and its summary arrive at a greater understanding of the actual directions NPR developers have chosen to explore, as well as a fuller definition of NPR. Several tendencies are revealed, the most notable of which is that developers often assume NPR to be synonymous with natural media simulation. This is a forgivable stance to adopt and follows Negroponte's maxim that every new medium must go through a phase of imitating a predecessor. However, it gives rise to three main objections: Firstly, the air of "phoniness" surrounding natural media simulation causes problems for many artists and designers. Secondly, is the suspicion that the goal of convincingly imitating natural media drawings and painting can never be achieved, so ought not be attempted. Thirdly is the feeling that this line of approach is fundamentally reactionary and over-limiting and that many new styles could be developed if this unnecessary cultural baggage were discarded.

A definition of NPR was forthcoming; that while PR is wholly concerned with the credible projection of 3-D artefacts into 2-D artefacts – so that everything seen in the image is supposed to "exist" in 3-D also, NPR is much more concerned with the image plane in isolation from the 3-D scene it represents, and that NPR techniques employ a tendency to impose 2-D artefacts in the image plane that clearly do not exist in world-space. These artefacts tend to be things such as marks, hatches and drawn edges.

In order to redress the three above objections the thesis argues (Chapter 5) that the trajectory of natural media simulation ought to be abandoned. This can be done by purging from future development certain strategies currently used in several systems. Specifically,

systems must abandon all attempts to convince that the images generated by such systems really have been, at some time, a natural media image (such as faking the impasto surface of oil painting, or the action of gravity on paint). Other representational devices, while initially seen in natural media, can be used with seeming impunity as they transcend the specifics of natural media (such as colour, wobbliness, ambiguity) . These latter aspects are unproblematic and can be maintained within future development. This position is in direct opposition to the one adopted at the beginning of the investigation; my working title for the PhD program was "Rendering Strategies that Model Painterly Marks" which admits the intention of natural media simulation. This shift in attitude – from one which thought the project was very much to do with painting and drawing, to one which regards the similarities as only incidental – has been a crucial and liberating one.

The thesis further calls for an improvement in the articulating power of current systems (Chapter 4) before they can claim to make "significant" images; i.e. images that embody new levels of meaning rather than simply presenting an optical distortion of a previously photorealistic image. In order to do this there is a need for systems to possess more information about scene they wish to depict, such as material properties, geometric information and the ability to distinguish one part of the scene from another. The prototype system "Magic Painter" (Chapter 8) while generating pleasing imagery, is no more sophisticated than a host of now popular systems, such as Fractal Painter or certain Adobe Photoshop plug-in modules, due to its reliance on colour information alone. However, it has been very valuable for exploring the dynamics of human-computer interaction - and indicated that the final solution might be somewhere between automatic rendering and interactive painting.

The discussions with designers and architects (Chapter 7) revealed one other major consideration - none of the developments of PR technologies enjoyed by practitioners should be excluded from an NPR system, for this reason NPR could be thought of a super-set of PR instead of an alternative.

All these points have been borne in mind throughout the development of the final system. Natural media simulation *per se* has been abandoned in favour of simple, well known mark-making techniques. While these mark-making techniques may bear a superficial resemblance to traditional media this is not their primary intention. Many of the techniques employed, such as the applying of cut-out shapes to the scene, bear more resemblance to computer-painting techniques than to traditional media. 2-D and 3-D



hatching, while being first used by engravers, is a practical way of simultaneously half-toning and delineating form in any medium.

Following a recommendation made in Chapter 4, the Expressive Marks rendering system (Section 8.4) has access to the geometric properties of the scene. However, in order to completely fulfil the recommendations of Chapter 4, and because of other technical limitations, this vector-based approach underwent complete reworking for the final Piranesi System (Chapter 11). This resulted in the implementing of the Enriched Pixel Image Format (Chapter 9). The Piranesi system is able to access the scene's colour, tone, geometry, material types and is able to distinguish between one part of the scene and another with the speed necessary for interactive techniques. Following the recommendations made in Chapter 7, that NPR is a super-set of PR, the system is able to achieve photorealistic effects – these may be completely independent of the source RGB image, but because the EPix format is manufactured by more traditional PR techniques, the Piranesi system enjoys all the PR techniques available to the exporting system.

The adoption of object-oriented techniques was also significant, as this led the way to generate a modular approach to rendering, where distinct plug-in modules are used to perform separate rendering tasks, rather than attempting to write a generalised suite of algorithms to cope with all cases. This has proved to be crucial for extensibility and flexibility, and has enabled the writing of many complex rendering styles, some of which are totally new. These can be easily added to the system, thereby further addressing the recommendations made in Chapter 4, calling for a general improvement in the sophistication of current NPR techniques.

The Piranesi System addresses many of the points made in the wish list and requirements specification (Section 7.5). In its current form the system :-

- is able to automatically render 3-D models or post-process images with no apparent distinction between the two processes.
- gives the user the ability to author new rendering styles by combining existent rendering devices or sub-classing to derive totally new styles.
- is able to apply these styles to the image with real-time interactive speeds.

- can generate images through a range of interactive or automatic techniques.
- has the ability to save and reload an image for further work.
- has the ability to introduce new levels of detail into the 3-D scene via interactive methods that would be impossible or difficult to model using conventional 3-D techniques.

The images made in the process of this investigation form the most important result of the investigation. Many hundreds have been generated, usually showing an individual technique in isolation. More recent image show combinations of techniques, such as the use of several NPR techniques together to make "neo-classical" drawings (Figure 37), or the use of overlaying PR texturing techniques using interactive painting method discussed in Section 7.2 and partially explored in the Magic Painter system(Figure 38).

A large portfolio of images is available from the author on request.

# Bibliography

- Aldus Gallery Effects. (1992). *Aldus Gallery Effects* Seattle: Aldus Corporation.  
Plug-in modules for other paint programs.
- Aono, M., & Kunji, T. L. (1984). *Botanical Tree Image Generation*. IEEE Computer Graphics Applications, Vol. 4 (5) (pp10-33).
- Armstrong, C (1887). *Cusack's Shading*. Bishopsgate Press, London.
- Ayer, F. C. (1916). *The Psychology of Drawing with special reference to laborotary teaching*. Unpublished doctoral dissertation, Baltimore University, Baltimore.
- van Bakergen, W. D. & Obata, G. (1992). *Free-hand Plotting - is it Live or is it Digital?* in G.N. Schmitt (ed.) , CAD Futures 91, Vieweg, Wiesbaden (pp567-582).
- Barnsley, M (1988). *Harnessing Chaos for Image Synthesis*. Computer Graphics Vol. 22, ACM Press (pp141-150).
- Barnsley, M., & Hurd, P. (1992). *Fractal Image Compression*. AK Peters, Wellesley MA.
- Barthes, R. (1980). *Camera Lucida*. Jonathan Cape, London.
- Barthes, R. (1977). *The Photographic Message*. Image-Music-Text. Flamingo Press, London.
- Baudrillard, J. (1989). *The Evil Dacmon of Images*. Power Inst for Fine Arts, Sydney (pp28-30).
- Baudrillard, J. (1983). *Simulations*. Semiotext(e), New York (pp10-44).
- Baudrillard, J. (1993). *Xerox and Infinity* , The Transparency of Evil. Verso Press, London.
- Becket, W., & Badler, I. (1990). *Imperfection for Realistic Image Synthesis*. Visualisation and Computer Animation Vol. 1, Wiley, Chichester UK(pp26-32).
- Bergin, V (1989). *Barthes, Diderot, Vertigo* , The End of Art Criticism. Verso, London.
- Beyer, H., & Streilein, A. (1991). *Data Generation for CAAD with Digital Photogrammetry* in G.N. Schmitt (ed.) , CAD Futures 91, Vieweg, Wiesbaden (pp539-542).
- Boden, M. (ed) (1990). *The Philosophy of Artificial Intelligence*. Oxford University Press, New York.
- Bouknight, J. (1970, September). *A Procedure for Generation of 3-D Half-toned CG Presentations*. Communications of the ACM Vol. 13, ACM Press (pp527-536).
- Bresenham, J. E. (1965). *Algorithm for Computer Control of a Digital Plotter*. IBM Systems Jounal Vol. 4, IBM Press, New York (pp25-30)

- Carol, D. (1987). *Paraesthetics*. Methuen, London.
- Carpenter, L. (1984). *The A-Buffer: an Anti-aliased Hidden Surface Method*. Computer Graphics Vol 18, ACM Press (pp 103-8).
- Catmull, J. (1974, December). *A Subdivision Algorithm for CG Display of Curved Surfaces*. Published doctoral dissertation, Computer Science Dept, University of Utah, USA.
- Chamberlain, H. (1990). *Musical Applications of Microprocessors*. Hayden, London (pp 90-91).
- Cirker, B. (1962). *Woodcuts by Thomas Bewick and his School*. Dover Pictorial Archives, New York.
- Cockshott, T. (1992). *Modelling the Texture of Paint*. Eurographics 92 Proceedings CGF Vol. 11, Blackwell (pp C217-C226).
- Cockshott, T. (1991). *Wet and Sticky: a novel model for computer-based painting*. Unpublished doctoral dissertation, Glasgow University, Glasgow.
- Crimp, D. (1985). *On The Museum's Ruin*. in Postmodern Culture. ed Hal Foster. Pluto Press, London.
- Deeney, S. (1988). *On the Road with Amazing Array*. Computer Images, May/June 1988 (pp14-16).
- Durisch, P, & Anderheggen (1991). *Leaving The Planar Univers*. Proceedings in G.N. Schmitt (ed.) , CAD Futures 91, Vieweg, Wiesbaden (pp521-534).
- Foley, J., & vanDam, A. (1990). *Fundamentals of Interactive Computer Graphics (Second Edition)*. Addison Wesley , Reading MA.
- Fractal Design(1991). *Fractal Design Painter* (software) Fractal Design Corp. Aptos CA.
- Gaur, A. (1987). *A History of Writing*. The British Library, London.
- Gill, R. W(1976). *Rendering With Pen and Ink*. Thames and Hudson, London.
- Glazzard, N. (1993, April). [Letter to Simon Schofield]. Private.
- Gombrich, E. (1960a). *Formula and Experience*. Art and Illusion. Phaidon, London (pp 124 -127).
- Gombrich, E. (1960b). *From Light to Paint*. Art and Illusion. Phaidon, London (pp29-54).
- Gombrich, E. (1970). *Meditations on a Hobby-horse*. Phaidon, London.
- Gombrich, E. (1990) *The Story of Art*. Phaidon, London (pp 416-7)
- Goodman, S., & Porter, T. (1991). *Design Drawing Techniques for Architects, Graphic Designers and Artists*. Butterworth Architecture, Oxford.

- Haeberli, P. (1990, August). *Painting by Numbers: Abstract Image Representation*. Computer Graphics Vol. 24, ACM Press (pp207-214).
- Hare, J. (1991, October). [Letter to S.Schofield]. Private. (on Softening computer images through repeated dithering).
- Heller, D. (1991). *Motif Programming Manual*. O'Reilly and Assoc. Sabastopol CA.
- Herberts, K (1959). *Artists' Techniques*. Thames and Hudson, London.
- Holladay, T. M. (1980). *An Optimal Algorithm for halftone Generation for Displays and Hard Copies*. Proceedings of the Society for Information Display Vol. 21 (pp 185-192).
- ImageWare. (1989). *Image Paint System* (software), Image Ware Systems, Toronto.
- Irvin, W. M. (1969). *The Microstructure of Various Printed Media*. Prints and Visual Communications. MIT Press. Boston.
- Jones, B. (1988). *Imitation and Representation of Realities*. Leonardo. SIGGRAPH '89 Art Show Supplementary Issue. Pergamon Press (pp31-38).
- Kajiya, J. (1989, August). *Rendering Fur with Three-Dimensional Textures*. Computer Graphics Vol. 23, ACM Press (pp271-280).
- Kandinsky, W (1947). *Point and Line to Plane*. Solomon Guggenheim, New York.
- KATI. (1993). *The KATI Renderer* (Beta Version). 5D Inc. London.
- Kernighan, B., & Ritchie, D.(1988). *The C Programming Language*. Prentice-Hall, New Jersey.
- Kirsch J & R. (1988). *Anatomy of Painting Style*. Leonardo Vol. 21, (pp 437-444).
- Klee, P. (1953). *The Pedagogical Sketchbooks*, Solomon Guggenheim, New York.
- Knuth, D. E. (1981). *The Art of Computer Programming: Semi-numerical Algorithms*. Addison-Wesley, New Jersey (pp 379 - 430).
- Koutamanis, A. (1993). *The Future of Visual Design Representations in Architecture*. Automation in Construction Vol. 1, Elsevier (pp 47-56).
- Leyotard, J. F. (1984). *The Postmodern Condition*. Manchester University Press, UK.
- Lindenmayer, A. (1968). *Mathematical Models for Cellular Interactions in Development, Parts 1 and 2*. Journal of Theoretical Biology Vol. 18, (pp 280-315).
- Loyal, T (1983). *Picture of Reality*. BFI press, London.



- Nagy, M (1965). *Painting-Photography-Film*. Studio Vista, London.
- Mezei, L. (1974). *Simulation of Patterns of Nature by Computer Graphics*. Information Processing 74, (pp 861-865).
- Mitchell, W. T. (1993). *The Reconfigured Eye*. MIT Press, Cambridge MA.
- Model (1993). *Model System Program (1.0)*. Wiegand, T., & Rees, J. Martin Centre, Cambridge UK.
- Monet (1993). *Monet Paint Program (2.0)*. Delta Tao, USA.
- Morton, M. (1990). *A Digital Dissolve Effect*. in Graphics Gems 1 (ed) Glassner, A. Academic Press, London (pp221-232).
- Neider, J. & Tillman, C S. (1992). *ImageVision Library Programming Guide*. Silicon Graphics, Mountain View CA.
- Newell, M E. (1972). *A Solution to the Hidden Surface Problem*. Proceedings of the ACM National Conf 1972, ACM Press (pp 443-450).
- Nochlin, L. (1971). *The Nature of Realism in Realism*. Penguin Books, London.
- Nye, A. (1990). *The XLib programming Manual Vol 1, Version 2*. O'Reilly and Assoc, Sebastopol CA (pp 339-348).
- Peach, D. R. (1985). *Solid Texturing of Complex Surfaces*. Computer Graphics Vol. 20, ACM Press (pp 279-286).
- Perlin, K. (1989, July). *Hypertexture*. Computer Graphics Vol. 23, ACM Press (pp253-262).
- Perlin, K. (1985). *An Image Synthesiser*. Computer Graphics Vol. 19, ACM Press (pp287-296).
- Photoshop (1993). *Photoshop Paint Program (2.5)*. Adobe Systems Inc. Mount View CA.
- Posch, K. C., & Fellner W D. (1989, January). *The Circular Brush Algorithm*. ACM TOG Vol. 1, ACM Press (pp 1- 24).
- SIGGRAPH (1990). *Computer Graphics*, Vol. 24, ACM Press (pp 197-223).
- Reeves, W. (1983). *Particle Systems - A Technique for Modelling Fuzzy Objects*. Computer Graphics Vol. 17, ACM Press (pp 359-279).
- Rewald, J. (1980). *The History of Impressionism*. Secker and Warburg, London (p 338).
- Rheingold, H. (1991). *Virtual Reality*. Secker and Warburg, London.
- Richens, P. (1993). *Does Knowledge Really Help? CAD Research at the Martin Centre in*

Knowledge-Based Computer Aided Design (ed) G. Carrara & Y.E.Kalay, Elsevier Amsterdam (pp305-25).

Rosenberg, H. (1969). *Signs and the Mythic Act*. Artworks and Packages. Phoenix Press, New York.

Saito, T., & Takahashi, T. (1990, August). *Comprehensible Rendering of 3-D Shapes*. Computer Graphics Vol. 24, ACM Press (pp 197-206).

Sasada, T. T. (1987). *Drawing Natural Scenery by Computer Graphics*. Computer Aided Design Vol. 19, (pp 212-218).

Schofield, S. (1989, September). *Rendering and Post-Processing in Painterly Styles*. Unpublished MA dissertation, Middlesex University, London.

Schofield, S. (1991, November). *A New Method of Computer Image Generation - Rendering Strategies that Model Painterly Marks Transfer* report MPhil/PhD, Middlesex University, London.

Schumacher, D (1990). *One to One Pixel Transforms* in Graphics Gems I. (ed) . Glassner, A. Academic Press, London.

Shamos, I., & Preparata, F (1985). *Computational Geometry*. Springer Verlag, (pp 204-211).

Sontag, S. (1977). *On Photography*. Penguin, London.

Sorokin, P. A. (1992). *The Crisis of Our Age*. Oneworld Press, Oxford.

Stiney, G., & Mitchell, W. (1978). *The Palladian Grammar*. Environment and Planning B Vol. 5, Pion Press, London (pp 227-238).

Strassman, S. (1986). *Hairy Brushes*. Computer Graphics Vol. 20, ACM Press (pp 225-231).

Strauss, P., & Carey, R. (1992). *An Object-Oriented 3-D Graphics Toolkit*. Computer Graphics Vol. 26, ACM Press. (pp 341-349).

Stroustrup, B. (1991). *The C++ Programming Language (2nd edition)*. Addison-Wesley, New Jersey.

Sze, M. (1977). *The Mustard-seed Garden Manual of Painting*. Princeton University Press.

Tagg, J. (1988). *Power and Photography*. The Burden of Representation. Macmillan Press, London.

Taylor, B. (1987). *Modernism, Postmodernism, Realism*. Winchester School of Art Press.

Ultra Paint. (1990). *UltraPaint Paint System* : Denaba Systems Inc. CA.

Upstill, S. (1989). *The Renderman Companion*. Addison Wesley, New Jersey.

- van Berkal, P. (1989, August). *Strokes Interpreted Animation Sequence*. European Computer Graphics Forum, (8), Elsevier (pp35-47).
- Watanabe, Y. et al. (1989). *Drawing Human Hair with the WISP Model*. New Advances in Computer Graphics. Springer-Verlag, New York.
- Weiler, K., & Atherton, P. (1977). *Hidden Surface Removal using Polygon Area Sorting*. Computer Graphics, Vol 11. (2), ACM press(pp 214-222).
- Weisenbaum, J (1976). *Computer Power and Human Reason*. WH Freeman, San Francisco.
- Werneck, J. (1992). *IRIS Inventor Programming Guide vols 1 and 2*. Silicon Graphics Inc, CA.
- Whitted, T. (1980, June). *An Improved Illumination Model for Shaded Display*. Communications of the ACM Vol. 23, ACM Press (pp 343-349).
- Wilton-Ely, J. (1978). *The Mind and Art of Giovanni Battista Piranesi*. Thames and Hudson, London.
- Wright, R. (1989). *The Image in Art and Computer Art*. Leonardo. SIGGRAPH '89 Art Show Supplementary Issue, Pergamon Press (pp49-53).
- Wyvill, B. (1986). *Animating Soft Objects*. The Visual Computer Vol. 2, Computer Graphics Society, Berlin (pp227-242).
- XDesigner (1992). *XDesigner User Manual* Imperial Software Technology, London.
- Yessios, C. I. (1979). *Computer Drafting of Stones, Wood, Plant and Ground Material*. Computer Graphics, Vol 13, ACM Press (pp 190-198).